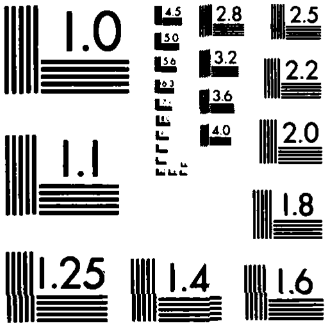
AD-A135 837   A USER'S MANUAL FOR ELECTROMAGNETIC SURFACE PATCH (ESP)   1/4
CODE VERSION II P..(U) OHIO STATE UNIV COLUMBUS
ELECTROSCIENCE LAB E H NEWMAN ET AL. SEP 83
UNCLASSIFIED   ESL-712692-3 N00014-78-C-0049   F/G 20/14   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

OSU

The Ohio State University

A USER'S MANUAL FOR ELECTROMAGNETIC SURFACE PATCH (ESP) CODE: VERSION II - POLYGONAL PLATES AND WIRES

E.H. Newman
P. Alexandropoulos

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report 712692-3
Contract No. N00014-78-C-0049
September 1983

Department of the Navy
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

50272-101

REPORT DOCUMENTATION PAGE

1. REPORT NO.

2.

3. Recipient's Accession No.
AD-A135 831

4. Title and Subtitle
A USER'S MANUAL FOR ELECTROMAGNETIC SURFACE PATCH (ESP) CODE: VERSION II - POLYGONAL PLATES AND WIRES

5. Report Date
September 1983

6.

7. Author(s)
E.H. Newman and P. Alexandropoulos

8. Performing Organization Rept. No.
ESL 712692-3

9. Performing Organization Name and Address
The Ohio State University ElectroScience Laboratory
Department of Electrical Engineering
Columbus, Ohio 43212

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No
(C)
(G) N00014-78-C-0049

12. Sponsoring Organization Name and Address
Department of the Navy
Office of Naval Research
Arlington, Virginia 22217

13. Type of Report & Period Covered
Technical Report

14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

This report serves as a user's manual for the Electromagnetic Surface Patch (ESP) Code. This code is a method of moments solution for interconnections of thin wires and polygonal plates. The code can compute currents, input impedance, efficiency, mutual coupling, and far-zone radiation and scattering patterns. In addition to describing the code input and output, the use of the code is illustrated by simple examples. Subroutine descriptions are also given.

17. Document Analysis a. Descriptors

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement
DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

19. Security Class (This Report)
Unclassified

20. Security Class (This Page)
Unclassified

21. No of Pages
302

22. Price

(See ANSI-Z39.18)

See Instructions on Reverse

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

This report deals with the development of a computer program based on the Moment Method (MM) solution for antenna and scattering problems. The program can solve for geometries consisting of polygonal plates, wires, wire/plate junctions and multiplate junctions. The advantages of the program are accuracy, versatility and the flexibility in the input of the problem geometry. It can calculate far-zone radiation and scattering patterns as well as antenna input impedances and efficiencies. The disadvatages of the program are the limitation to geometries not large in terms of wavelength and the lack of analytical results which can provide physical insight into the problem.

The first implementation of the Moment Method solution into a useful computer program involved the thin-wire formulation [1,2]. These programs gave good results for most wire antennas and by forming wire grid models of solid surfaces reasonable results were obtained [3]. The wire grid approach was limited to solid surfaces whose dimensions in terms of wavelength were very small. Also it did not give accurate results for near-zone parameters such as current distributions and input impedances.

The next development was the use of surface patches for the modeling of three dimensional surfaces. Surface patches give a much more accurate approximation to the currents on a three-dimensional surface and require less unknowns than the wire grid model. Oshiro [4] used pulse basis functions and point matching to solve the Magnetic Field Integral Equation (MFIE) for various three dimensional surfaces. Albertsen, et al. [5], used pulse test modes and the MFIE formulation to model wire, plates and wire/plate attachments. However, their solution was limited to closed surfaces since the MFIE applies to that type of surface only. Based on the MFIE formulation, the Numerical Electromagnetic Code (NEC) was developed by Burke and Poggio [6] to solve for geometries consisting of wires and closed surfaces. Wang and Richmond [7] used piecewise sinusoidal (PWS) rectangular surface patches to model rectangular plates and wires. Surface patch models using triangular patches and pulse test modes have been developed by Glisson [8] and can handle wires, plates and wire/plate junctions.

This report incorporates the work of Newman and Pozar [9,10] and Tulyathan [11].

Chapter Two gives a brief review of the MM solution for electromagnetic scattering and radiation problems, based on the Reaction Integral Equation (RIE) [12]. The wire, plate, attachment and overlap modes are defined. Various special methods for calculating the impedance matrix effieciently and accurately are also discussed.

Chapter Three describes the READ input statements of the main program. For every READ statement a detailed explanation of all the parameters introduced is given. Subroutine WGEOM is described and two examples illustrating the methodology of creating and the advantages of using such a subroutine are given. Finally, several design examples are given for a better understanding of the input parameters.

Chapter Four contains descriptions of every subroutine called by the main program. Excluded are the subroutines documented in [1] and also several subroutines that are system dependent. For every subroutine the following format is used:

1. A brief statement of its purpose.
2. The general calling form.
3. Detailed definition of every input and output parameter.
4. A brief outline of the subroutine's inner workings, unless self evident.

Chapter Five is the summary.

CHAPTER II

FORMULATION OF THE MOMENT METHOD SOLUTION

A. THEORY

This chapter gives a brief outline of the solution of the electromagnetic scattering or antenna problem by the Method of Moments (MM). A description of the expansion (basis) functions is given, along with a discussion of various impedance calculation methods.

Consider an arbitrarily shaped scatterer in a homogeneous medium. Let S represent the surface of the scatterer and $\hat{n}$ the unit outward normal to the surface. ($\underline{J}i$,$\underline{M}i$) is an impressed source which radiates fields ($\underline{E}i$,$\underline{H}i$) in free space and fields ($\underline{E}$,$\underline{H}$) in the presence of the scatterer (see Figure 2-1).

From Schelkunoff's surface equivalence theorem [13] the field interior to the surface S will vanish without changing the exterior field (E,H) if one introduces the following electric and magnetic surface current densities on surface S:

$$\underline{J}s = \hat{n} \times \underline{H} \tag{2.1}$$

$$\underline{M}s = \underline{E} \times \hat{n} \tag{2.2}$$

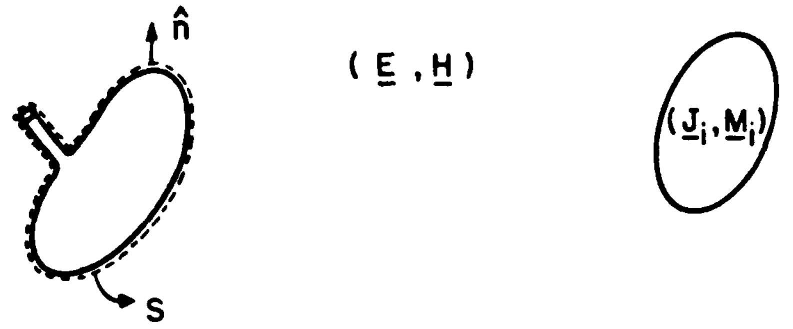

$\hat{n}$
$(\underline{E}, \underline{H})$
$(\underline{J}_i, \underline{M}_i)$
S

Figure 2-1. Source $(\underline{J}_i, \underline{M}_i)$ radiates fields $(\underline{E}, \underline{H})$ in the presence of the scatterer.

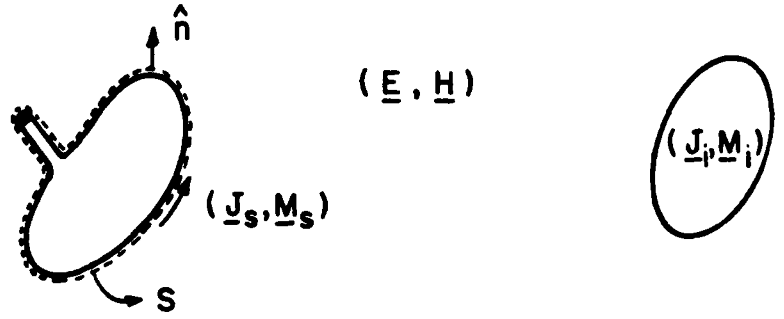

$\hat{n}$
$(\underline{E}, \underline{H})$
$(\underline{J}_i, \underline{M}_i)$
$(\underline{J}_s, \underline{M}_s)$
S

Figure 2-2. Surface currents $(\underline{J}_s, \underline{M}_s)$ placed on the surface of the scatterer do not change the exterior fields $(\underline{E}, \underline{H})$.

where $\underline{E}$ and $\underline{H}$ are the electric and magnetic fields, respectively, on the surface S. These surface current densities radiate, in the ambient medium, the scattered fields ($\underline{Es}$,$\underline{Hs}$) (see Figure 2-2) which are defined by:

$$\underline{Es} = \underline{E} - \underline{Ei} \tag{2.3}$$

$$\underline{Hs} = \underline{H} - \underline{Hi} . \tag{2.4}$$

If one place a test source ($\underline{Jm}$,$\underline{Mm}$) in the volume interior to surface S its reaction with the sources ($\underline{Ji}$,$\underline{Mi}$) and ($\underline{Js}$,$\underline{Ms}$) will be zero since the field interior to surface S is zero, i.e.,

$$\iint_s (\underline{Jm}.\underline{Es} - \underline{Mm}.\underline{Hs})ds = - \iint_s (\underline{Jm}.\underline{Ei} - \underline{Mm}.\underline{Hi})ds. \tag{2.5}$$

Using the reciprocity theorem, Equation (2.5) can be written as

$$\iint_s (\underline{Js}.\underline{Em} - \underline{Ms}.\underline{Hm})ds + \iiint_V (\underline{Ji}.\underline{Em} - \underline{Mi}.\underline{Hm})dv = 0 \tag{2.6}$$

where V is the volume occupied by source ($\underline{Ji}$,$\underline{Mi}$).

This is the Reaction Integral Equation (RIE). If one uses a set of electric test sources, the RIE reduces to the Electric Field Integral Equation (EFIE). If a set of magnetic test sources is used, the RIE reduces to the Magnetic Field Integral Equation (MFIE). The EFIE is used in this work since it applies to both closed and open surfaces while the MFIE applies to closed surfaces only. Thus one can take $\underline{Mm} = \underline{0}$. Also, perfect conductivity is assumed for the scatterer surface and thus $\underline{Ms} = \underline{0}$.

All of the above analysis was based on the assumption that the surfaces in consideration are closed. However it can be shown that the analysis is valid for open surfaces such as surface plates. This is very important since surface plate modeling is the core of the Electromagnetic Surface Patch Code (ESP). The plates used in the ESP code are fictitious in the sense that they have zero thickness. In general different currents exist on the top and bottom surfaces of a real plate. As the thickness of the plate goes to zero the fields radiated by the top and bottom currents become equivalent to the field radiated by a single current located at the center of the plate. This current, which is $\underline{J}s$ of Equation (2.6), is the vector sum of the top and bottom surface currents of the plate [14].

$\underline{J}s$ represents the unknown current on the surface of the scatterer. The Moment Method solution begins by expanding $\underline{J}s$ in terms of N expansion (basis) functions $\underline{F}n$, i.e.,

$$\underline{J}s = \sum_{n=1}^{N} In\underline{F}n \quad . \tag{2.7}$$

Substituting $\underline{J}s$ from Equation (2.7) into Equation (2.6) one obtains:

$$\sum_{n=1}^{N} InZmn = Vm \quad ; \quad m = 1,N \tag{2.8}$$

where

$$Zmn = - \iint_{n} \underline{E}m.\underline{F}nds \quad , \tag{2.9}$$

$$Vm = \iiint_{V} \underline{J}m.\underline{E}idv \quad . \tag{2.10}$$

The integral in Equation (2.9) is over the surface of the n-th expansion mode while the one in Equation (2.10) is over the volume occupied by source ($\underline{J}i,\underline{M}i$). Vm is called the excitation voltage. The detailed evaluations of Equations (2.9) and (2.10) are described in references [9,10].

B. EXPANSION MODES

Three types of basis functions (modes) are used in the moment method computer code, i.e., wire, surface patch and attachment dipole modes. With this choice of modes one can model geometries consisting of wires and/or polygonal plates or any geometry that can be approximated by wires and polygonal plates.

1. Wire Mode

The wire mode is the piecewise sinusoidal V-dipole consisting of two sinusoidal monopoles. Figure 2-3 shows a V-dipole with 180 degree interior angle [10]. The current on this dipole is given by

$$\underline{J}_s = \frac{\hat{z}}{2\pi a} \left[P_1 \frac{\sin k(z-z_1)}{\sin k(z_2-z_1)} + P_2 \frac{\sin k(z_3-z)}{\sin k(z_3-z_2)}\right] \quad , \qquad (2.11)$$

where

$$P_1 = \begin{cases} 1 & z_1<z<z_2 \\ 0 & \text{elsewhere} \end{cases}$$

$$P_2 = \begin{cases} 1 & z_2<z<z_3 \\ 0 & \text{elsewhere} \end{cases}$$

a = the wire radius and $k = 2\pi/\lambda$.

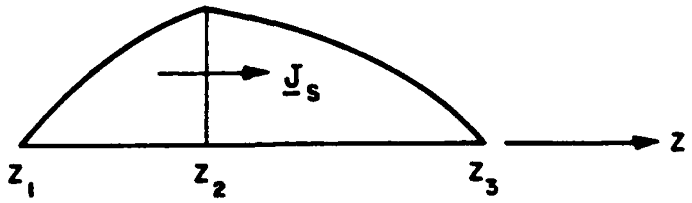


Figure 2-3. A wire PWS dipole mode.

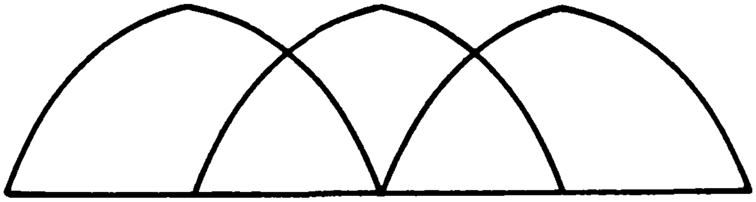


Figure 2-4. Array of overlapping PWS wire dipoles representing the current on a wire.

When wire modes are used to represent the current on a real wire they are placed in an overlapping array as shown in Figure 2-4. This ensures the continuity of current on the wire.

2. Surface Patch Mode

Two kinds of surface patch modes are used, i.e., a rectangular and a quadrilateral. The rectangular mode is a surface V-dipole consisting of two rectangular sinusoidal surface patch monopoles. A surface V-dipole with an interior angle of 180 degrees is shown in Figure 2-5 [10].

The current density on this dipole is given by

$$\underline{J}_s = \hat{z}\,\frac{kP_1 \sin k(z-z_1)\cos ky}{2\sin k(z_2-z_1)\sin kw} + \hat{z}\,\frac{kP_2 \sin k(z_3-z)\cos ky}{\sin k(z_3-z_2)\sin kw} \qquad (2.12)$$

where P1 and P2 are the unit pulse functions as described for the wire dipole.

Two orthogonal and overlapping arrays of rectangular surface patch modes are used to represent the current density on a rectangular plate as shown in Figure 2-6. Each arrow represents a V-dipole. This modal outlay ensures continuity of current on the surface of the plate and it makes the current density a two dimensional vector.

If the plate is not rectangular then quadrilateral V-dipole surface patches are used. A typical quadrilateral surface patch mode is shown in Figure 2-7.

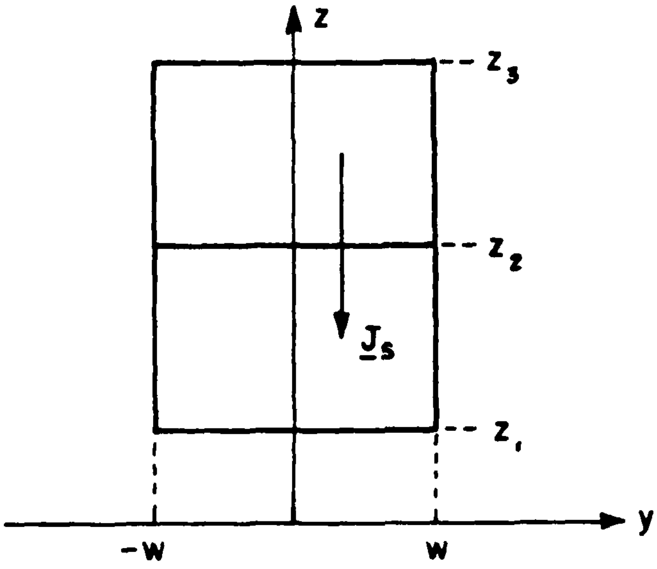


Figure 2-5. A PWS rectangular surface patch dipole mode.

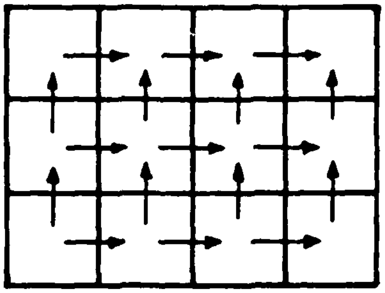


Figure 2-6. A two dimensional array of overlapping rectangular surface patch dipole modes representing the current density on a rectangular plate. The modes are represented by arrows.

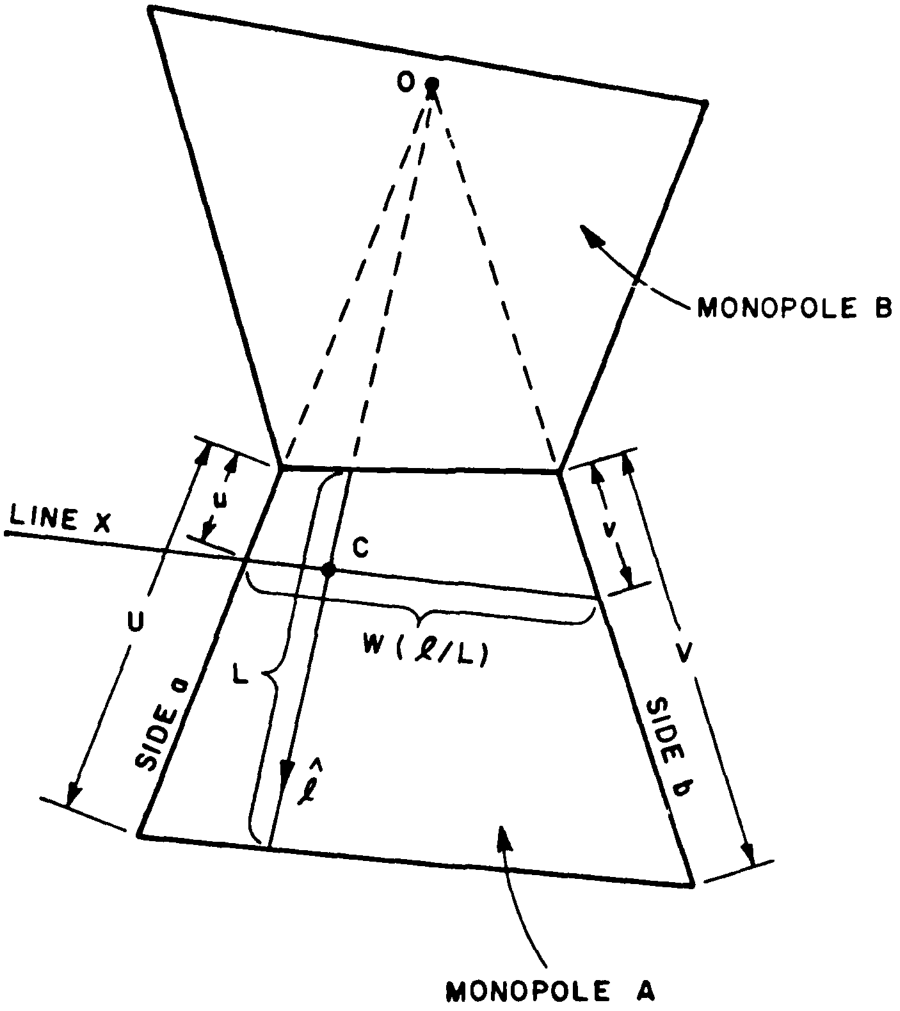


Figure 2-7. A quadrilateral surface patch dipole mode.

To describe the current density on the patch consider a point C interior to monopole A. Draw a line X intersecting sides "a" and "b" in such a way that u/U equals v/V. The intersection of sides a and b is point O. A line from point O to the end side of monopole A will be divided by line X such that $\ell/L=u/U=v/V$. L is the segment of the line drawn from O between the terminal and end side. The coordinate along segment L is ℓ ($\ell = 0$ at the terminal and $\ell = L$ at the end) and $W(\ell/L)$ is the length of the segment of line X between sides "a" and "b".

Now the current density on monopole A of the surface patch mode is

$$\underline{J}_{S_A} = c\,\hat{\ell}\,\frac{\sin k\ell}{\sin kL\; w(\ell/L)} \quad . \tag{2.13}$$

The constant C is chosen so that the current at the terminal side of monopole A is one ampere.

The density on monopole B is

$$\underline{J}_{S_B} = c'\,\hat{t}\,\frac{\sin kt}{\sin kT\; w(t/T)} \tag{2.14}$$

where t and T are defined the same way as ℓ and L were defined for monopole A. The constant C' is chosen so that the current at the terminal side of monopole B is equal to one ampere.

This surface patch mode is similar to the rectangular patch mode used by Newman and Pozar [9].

3. Attachment Mode

The attachment mode, shown in Figure 2-8(a), is used to model the wire/plate junction. Note that the wire is not necessarily always perpendicular to the disk. The attachment mode serves two purposes:

1. Ensures the continuity of current at the junction.
2. Ensures the proper ρ polarization and $1/\rho$ dependence of the current density at the junction.

It is composed of two monopoles, the wire monopole which is similar to the wire monopole described by Equation (2.11) and a disk monopole. The current density of the wire part is

$$\underline{J}_S^W = \frac{1}{2\pi a} \frac{\sin k(z_2 - z)}{\sin k z_2} \hat{z} \quad ; \quad 0 < z < z_2 \quad , \qquad (2.15)$$

while the current density on the disk is

$$\underline{J}_S^D = - \frac{\sin k(b - \rho)}{2\pi \rho \sin k(b - a)} \hat{\rho} \; ; \quad a < \rho < b \qquad (2.16)$$

a = radius of the wire and b = outer radius of disk. See Figure 2-8(b).

Note that the disk density at $\rho = a$ equals the wire density at $z = 0$, insuring continuity of current at the junction. Also, the density at the edge of the disk ($\rho = b$) is zero to maintain continuity of current on the plate where the disk is placed.

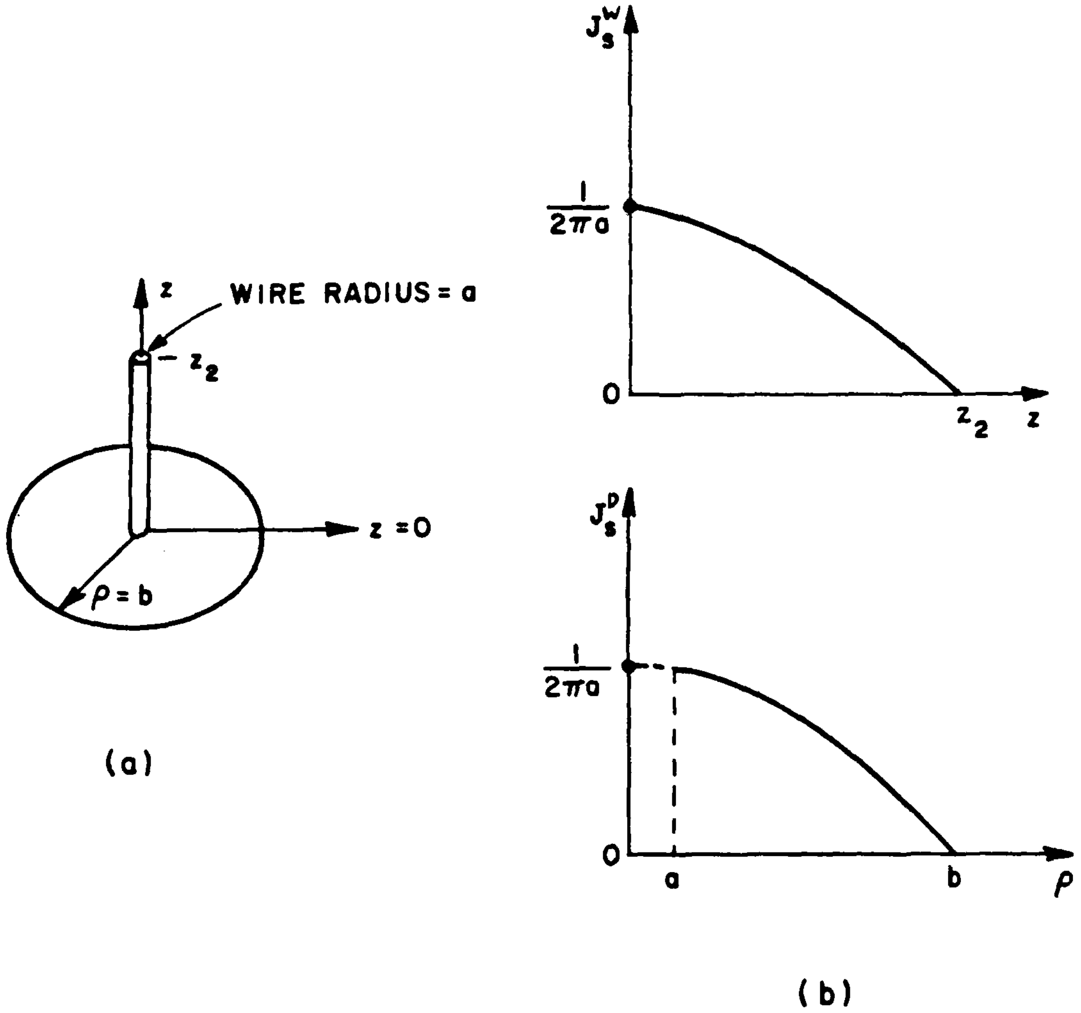


Figure 2-8. (a) Wire attachment dipole mode, and
(b) Current density on the wire monopole (top) and the disk monopole (bottom).

The attachment mode is placed directly over the surface of the plate at the wire/plate junction. The only restriction is that the wire/plate junction be at least 0.1λ away from all edges of the plate. Through convergence tests it was found that a good value for the outer radius b is 0.2λ.

4. Overlap Mode

The overlap modes are identical in mathematical description to the surface patch modes. They allow for the continuity of current at a plate-to-plate intersection. The edges of the overlap modes need not coincide with the edges of the surface patch modes on either plate. However, the closer they match the better the results seem to become. The program automatically searches for plates with touching sides and places the corresponding overlap modes.

C. TEST MODES

Normally the test modes used in the MM solution are identical to the expansion modes (Galerkin's method). This results in a symmetric impedance matrix and only its lower triangular part is evaluated.

Most of the computer CPU time is spent in calculating the elements of the impedance matrix. Substantial CPU time can be saved, without compromising the accuracy of the solution, by representing the test modes as single filaments. The endpoints of a filament are defined by the midpoints of the terminal and end sides of the surface patch it represents.

D. COMPUTATION OF THE IMPEDANCE ELEMENTS

Since there are three types of expansion and test modes the impedance matrix consists of nine types of impedance elements. This is shown graphically in Figure 2-9. It should be noted that the mutual impedance between two dipole modes is simply the sum of four monopole-to-monopole impedances.

Of the nine types of dipole-to-dipole impedances shown in Figure 2-9, five involve wire dipoles and require little computation time. Attachment-to-attachment impedances occur infrequently and thus do not require very much CPU time. Since most of the computation time is spent for surface patch-to-surface patch and disk-to-surface patch mutual impedances, the discussion that follows refers to those type of calculations.

A general impedance matrix entry Zmn is defined by Equation (2.9), or more explicitly

$$Z_{mn} = - \int_{e_1} \int_{e_2} \underline{E}_m(e_1,e_2) \cdot \underline{J}_n(e_1,e_2) \, de_1 de_2 \qquad (2.17)$$

where e_1,e_2 are independent coordinates on the surface of the n-th expansion mode, $\underline{J}_n$ is the current density of the n-th expansion mode and $\underline{E}_m$ is the electric field in free space of the m-th test mode. In particular, one finds that

$$\underline{E}_m(e_1,e_2) = \int_{t_1} \int_{t_2} \overline{\overline{G}}_0 \, (e_1,e_2;t_1,t_2) \cdot \underline{J}_m(t_1,t_2) dt_1 dt_2 \qquad (2.18)$$

$$\left[\begin{array}{c|c|c} {}^{W}/_{W} & {}^{W}/_{P} & {}^{W}/_{A} \\ \hline {}^{P}/_{W} & {}^{P}/_{P} & {}^{P}/_{A} \\ \hline {}^{A}/_{W} & {}^{A}/_{P} & {}^{A}/_{A} \end{array}\right]$$

W = WIRE
P = PLATE
A = ATTACHMENT

Figure 2-9. Symbolic representation of the nine different blocks of the impedance matrix.

where $\underline{J}_m$ is the current density of the m-th test mode, t_1 and t_2 are independent coordinates on the surface of the m-th test mode, and $\overline{\overline{G}}_0$ is the free space Dyadic Green's function. For a piecewise sinusoidal mode the field $\underline{E}_m$ is known in closed form for a wire [15] and surface patch [16] monopole and can be evaluated with a numerical integration for a disk monopole.

There is a way of avoiding the double integration of Equation (2.18) if we consider both the expansion and test mode as being made up of piecewise sinusoidal filaments. Then the impedance between a filament on the expansion mode (in the e_1 direction) and a filament on the test mode (in the t_1 direction) is given by (see Figure 2-10 for the filementary representation of two surface patch monopoles)

$$z_{mn} = - \int_{e_1} \underline{J}_n(e_1,e_2) \cdot \int_{t_1} \overline{\overline{G}}_0(e_1,e_2;t_1,t_2) \cdot \underline{J}_m(t_1,t_2)dt_1de_1 \quad . \tag{2.19}$$

This expression is known in closed form for piecewise sinusoids [17]. The total impedance between the expansion and the test monopole is

$$Z_{mn} = \int_{e_2}\int_{t_2} z_{mn}(e_2,t_2)de_2dt_2 \quad . \tag{2.20}$$

The evaluation of the double integral is done numerically in the code, usually using a Simpson's rule integration or Spline integration. The main advantage of using the second method (Equations 2.19 and 2.20) is that it results in a simpler computer program. In particular only one subroutine is needed to evaluate z_{mn} while for method one (Equation 2.17) three different subroutines are needed for calculating $\underline{E}_m$, the electric field of each type of test monopole. However, both methods are used for efficient computation of the impedance elements. Several examples of how each method is used are given below along with other special computation techniques.

1. Surface Patches Monopoles

A surface patch monopole is represented by NPT filaments (see Figure 2-10) and the mutual impedance between two surface patch monopoles is then the weighted sum of their mutual filament to filament impedances. The choice of NPT is important for the accuracy of the impedance matrix elements and the computation time required to evaluate those elements. Through extensive convergence tests it was found that NPT should be set based on DIST, where DIST is the center to center distance between two surface patch monopoles, as follows:

1. If $0 < \text{DIST} <= 0.25\lambda$ NPT = 8
2. If $0.25\lambda.$ $< \text{DIST} <= 0.35\lambda$ NPT = 4
3. If $0.35\lambda.$ $< \text{DIST} <= 0.6\lambda$ NPT = 2
4. If $\text{DIST} > 0.6\lambda$ NPT = 1.

The one-filament representation for each surface patch monopole can be quickly and easily modified to increase accuracy. If Z_{ff} is the filament-to-filament impedance then the variation due to the actual size and orientation of the n-th surface patch expansion monopole is taken into account by [11]

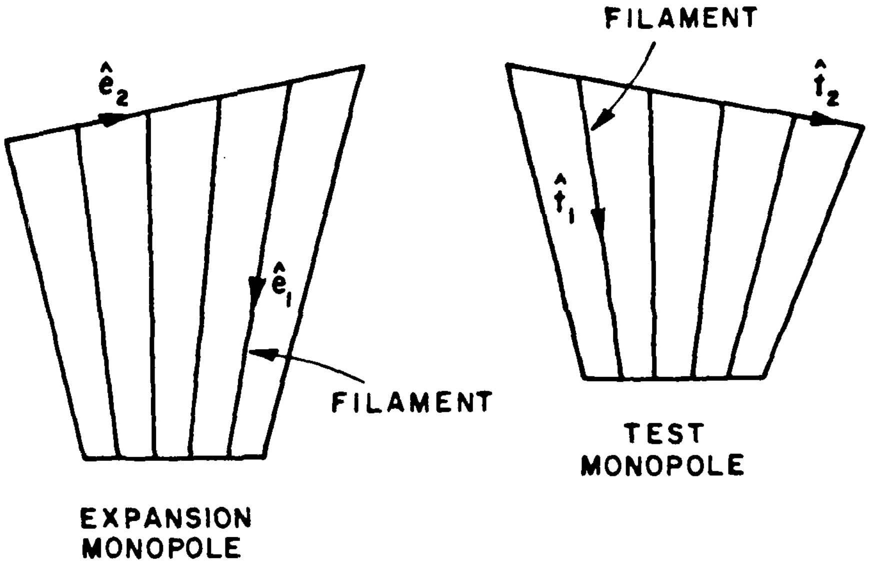


Figure 2-10. Filamentary representation of a test surface patch monopole and an expansion surface patch monopole.

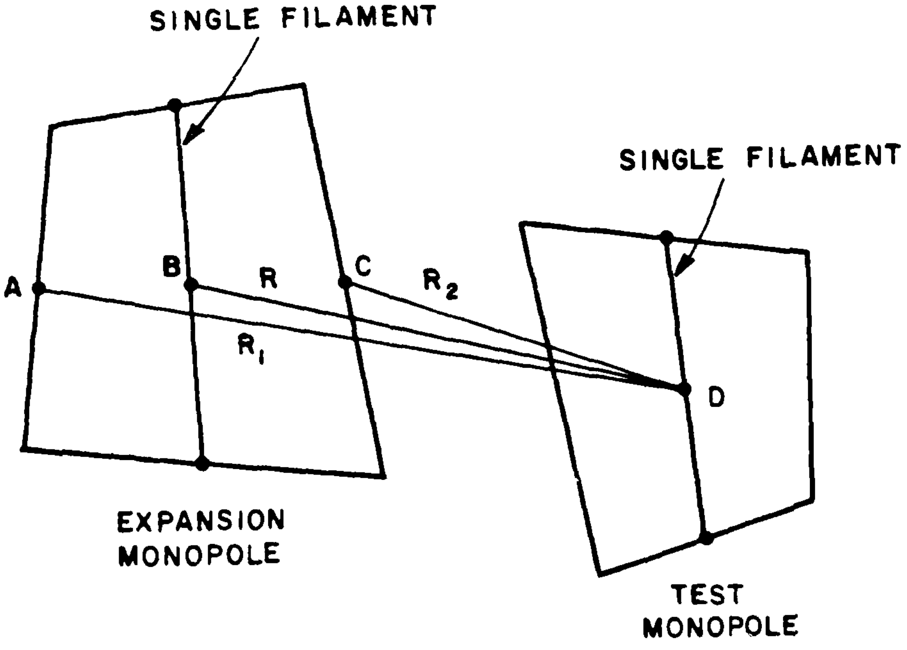


Figure 2-11. Single filament representation of two surface patch monopoles. Points A,B,C,D are the midpoints of their respective segments.

$$Z_{fn} = Z_{ff}\left(\frac{Re^{-jkR}}{R} \quad \frac{e^{-jkR_1}}{R_1} + \frac{4e^{-jkR}}{R} + \frac{e^{-jkR_2}}{R_2}\right) \qquad (2.21)$$

where R_1, R_2, R_3 are the average distances from the test filament to the closer, center and furthest edge, respectively of the expansion surface patch monopole (see Figure 2-11).

If the variation due to the m-th test monopole is taken into consideration then

$$Z_{m_n} = Z_{fn}\left(\frac{Re^{-jkR}}{R} \quad \frac{e^{-jkR_1}}{R_1} + \frac{4e^{-jkR}}{R} + \frac{e^{-jkR_2}}{R_2}\right) . \qquad (2.22)$$

2. Two Parallel Surface Patch Monopoles

Another advantage of the use of Equation (2.20) comes when calculating the mutual impedances between two rectangular surface patch monopoles which have current vectors parallel. This includes monopoles which have the vectors transverse to the current direction vectors parallel (see Figure 2-12). If each monopole is represented by M filaments, then M^2 filament-to-filament impedances need to be calculated. However, no more than 2M impedances are different and the rest can be evaluated from those 2M entries. This makes the computation time proportional to 2M instead of M^2.

3. Touching Surface Patch Monopoles

The integral of Equation (2.20) converges slowly when computing the mutual impedance between two touching monopoles. This is due to the

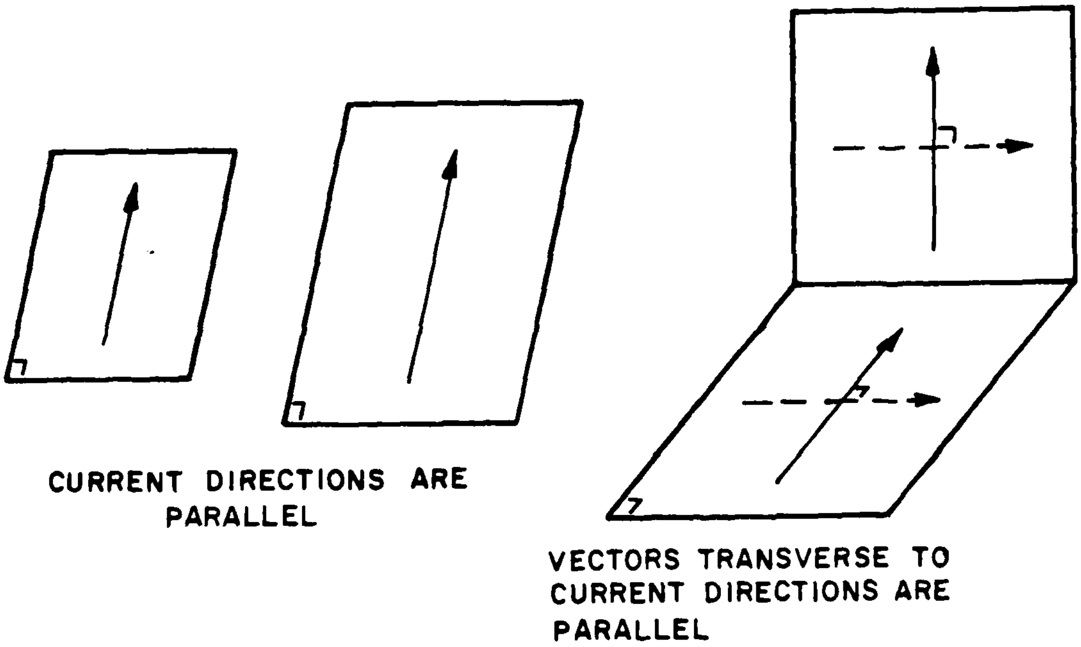

CURRENT DIRECTIONS ARE PARALLEL
VECTORS TRANSVERSE TO CURRENT DIRECTIONS ARE PARALLEL

Figure 2-12. Two cases of parallel rectangular surface patch monopoles.

fact that the imaginary part of the mutual impedance of two piecewise sinusoidals has a logrithmic singularity as the separation between the two filaments gets smaller [18]. For a small separation x the reactance between the two filaments can be written as

$$X(x) = C_1 + C_2 1_n(x) \quad . \tag{2.23}$$

The constants C_1 and C_2 can be evaluated and the ln singularity is integrated analytically. If Δ_x is a numerical integration interval then the equivalent value of the reactance at x = 0 is

$$X(0) = -X(\Delta x) + 2C_1 + 2C_2[1_n(\Delta x)-1] \quad , \tag{2.26}$$

where

$$C_2 = \frac{X(\Delta x) - X(\Delta x/2)}{1_n 2} \tag{2.24}$$

$$C_1 = X(\Delta x) - C_2 1_n(\Delta x) \tag{2.25}$$

X(0) is not the reactance at x = 0 (which would be infinite), but rather the value of X at zero that makes the numerical integration correct [10].

4. Toeplitz Properties

The impedance matrix for a single rectangular plate displays a great deal of toeplitz properties which can be used to reduce the computation time.

This property comes into play when computing the mutual impedance between modes on the same rectangular plate. Consider the typical modal layout for a 0.5λ by 1λ rectangular plate as shown in Figure 2-13.

Modes with the same current polarization are of equal size. In the example of Figure 2-13 modes 1, 2, 3, 4 are the same and modes 5, 6, 7, 8, 9, 10 are the same. It is obvious that $Z_{51} = -Z_{52}$, $Z_{58} = Z_{69} = Z_{710}$, $Z_{68} = -Z_{59}$, etc. This indicates that in general one does not need to calculate all of the mutual impedances between modes on the same rectangular plate. It is only necessary to compute the mutual impedances between the first mode in the 1-2 direction and all the modes on the plate and the mutual between the first mode in the 2-3 direction and all the 2-3 modes. This is shown graphically in Figure 2-14 where X's represent mutuals that are calculated and O's represent mutuals that can be obtained using the Toeplitz properties. If N12 is the number of modes in the 1-2 direction and N23 the number of modes in the 2-3 direction, then instead of calculating (N12 + N23)**2/2 + (N12 + N23)/2 mutual impedances one only has to calculate N12 + 2*N23.

5. Disk Parallel To Plate

Whenever a wire attachment is used substantial time can be saved in computing impedances between the disk monopole and plate modes on any plate parallel to the disk monopole. Here advantage is taken of the fact that the electric field of the disk part of the attachment mode has a radial component which is a function of the radial distance only. Using this property a table containing the values of E_ρ versus ρ is

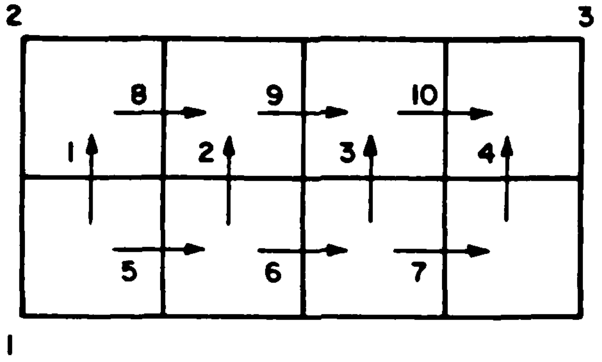


Figure 2-13. Model layout on a rectangular plate. The mode numbers are shown next to the arrays.

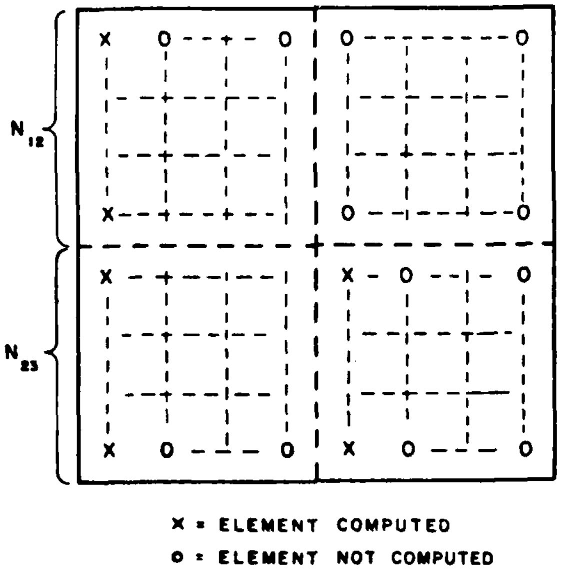


Figure 2-14. Symbolic representation of a P/P impedance matrix block. All entries in the block are the mutual impedances between modes on the same rectangular plate.

created and then used to extrapolate the value of E_m to be used in Equation (2.17). The integral in Equation (2.17) is actually evaluated numerically in the code as described in subroutines DSKTST and DSKTS2.

CHAPTER III

MAIN PROGRAM INPUTS

The inputs to the Electromagnetic Surface Patch Code (ESP) are explained below. They are used to describe to the program the detailed geometry of the problem and indicate the type of calculation desired. The input data can be broken into four major groups as follows:

1. Miscellaneous parameters.
2. Type of calculation desired.
3. Plate geometry.
4. Wire and attachment geometry.

The first three parts are always defined by an input file. The wire and attachment geometry can be defined either by an input file or by a FORTRAN subroutine called WGEOM. At first, the use of a seperate subroutine for describing the wire geometry might seem as an unnecessary complication. However, experience has shown that subroutine WGEOM is very useful for cases where the wire structure has a regular or periodic geometry or a shape that can be defined by an analytic expression. Examples are monopole and dipole antennas, loop antennas, helical antennas, log periodic antennas and arrays. For further explanation about the use of WGEOM see Section 3.

A. READ INPUT STATEMENTS

A description of every READ input statement is given below along with a definition of every parameter introduced. The fifteen READ input statements are shown in Figure 3-1. Also shown is some of the main program logic, indicating the order and number of times each READ is called. All READ input statements use a free format input on logical unit 5.

1. READ 1

The first READ input statement defines the following control parameters:

NGO = run indicator.

= 0 implies input and print out problem geometry and then stop, i.e., do not calculate any patterns or data. An NGO = 0 run should precede any pattern or data calculations. It gives the user the opportunity to review the accuracy of the problem geometry as defined by the input file.

= 1 implies go through the whole program, i.e. input the geometry and calculate the required patterns or data.

NPRINT = print indicator.

= 1 implies print wire and plate geometry.

= 2 implies print both the input parameters and the wire/plate geometry. Normally NPRINT = 2.

= 3 implies print nothing.

```
(1)-----READ(5,*)NGO,NPRINT,NRUNS,NWGS,IWR,IWRZT,INT,INTP,INTD,INWR,IRGM,
     1  IFIL
(2)-----READ(5,*)IFE,IPFE,NDFE,PHFE
(3)-----READ(5,*)IFA,IPFA,NDFA,THFA
(4)-----READ(5,*)ISE,IPSE,NDSE,PHSE,THIN,PHIN
(5)-----READ(5,*)ISA,IPSA,NDSA,THSA
        DO700NRUN=1,NRUNS
(6)-----READ(5,*)FMC,CMM,A
(7)-----READ(5,*)NPLTS
        IF(NPLTS.EQ.0)GOTO462
        DO464NPL=1,NPLTS
(8)-----READ(5,*)NCNRS(NPL),SEGM(NPL),IREC(NPL),IPN(NPL),IGS(NPL)
        DO466NCNR=1,NCNRS(NPL)
(9)-----READ(5,*)PCN(1,NCNR,NPL),PCN(2,NCNR,NPL),PCN(3,NCNR,NPL)
466     CONTINUE
464     CONTINUE
        DO 600 NWG=1,NWGS
(10)----READ(5,*)IWRZM,IRDZM
        IF(INWR.EQ.0)GOTO2773
        IF(IRGM.EQ.0)GOTO2800
(11)----READ(5,*)NM,NP,NAT,NFPT,NFS1,NFS2
        DO2810I=1,NP
(12)----READ(5,*)X(I),Y(I),Z(I)
2810    CONTINUE
        DO2820I=1,NM
(13)----READ(5,*)IA(I),IB(I)
2820    CONTINUE
        DO2830I=1,NFPT
(14)----IF(NFPT.GE.1)READ(5,*)IFM,IAB,VLG,ZL
2830    CONTINUE
        IF(NAT.EQ.0)GOTO2850
        DO2840I=1,NAT
(15)----READ(5,*)NAS,IAB,NPLA(I),VGA(I),ZLDA(I),BDSK(I)
2840    CONTINUE
        GOTO2850
2800    CALLWGEOM(IA,IB,X,Y,Z,NM,NP,NAT,NSA,NPLA,VGA,BDSK,
     2  ZLDA,NWG,VG,ZLD,WV,NFS1,NFS2)
2850    CALL SORT(IA,IB,I1,I2,I3,JA,JB,MD,ND,NM,NP,NWR,MAX,MIN,ICJ,INM)
2773    CONTINUE
                    .
                    .
                    .
*** MAIN BODY OF PROGRAM ***
                    .
                    .
                    .
600     CONTINUE
700     CONTINUE
```

Figure 3-1. The 15 READ input statements.

NRUNS = the number of runs to be made, i.e., the limit of the DO 700 loop in Figure 3-1.

NWGS = the number of wire geometries for each run, i.e., the limit of the 600 loop.

IWR = indicator for writing out the induced modal currents.

= 0 implies do not print the induced modal currents.

= 1 implies print the induced modal currents plus the detailed wire and plate modal geometry.

IWRZT = indicator for writing the impedance matrix on the output file.

= 0 implies do not write the impedance matrix on the output file.

= 1 implies write the impedance matrix on the output file.

INT = the number of Simpson's rule integration intervals used for the evaluation of the wire-to-wire impedances. INT is always an even integer, usually equal to 4.

= 0 implies the impedance calculations are to be done using the exact closed form expression. Self or overlapping wire impedances are always calculated by the closed form expression because it is more accurate than the numerical integration. However, the closed form expression is more time consuming than the INT = 4 numerical integration.

INTP = the number of Simpson's rule integration intervals used in integrating over the surface patch monopoles. INTP is always an even integer, typically chosen as 8.

INTD = the number of Simpson's rule integration intervals used in integrating over the disk monopoles. INTD is always an even integer, typically chosen as 18.

INWR = wire indicator.

= 0 implies geometry does not contain any wires.

= 1 implies geometry contains wires.

IRGM = indicator for choosing the method of defining the wire geometry.

= 0 implies the wire geometry is to be defined by subroutine WGEOM.

= 1 implies the wire geometry is to be read in via the input file.

IFIL = indicator for choosing the type of test plate modes.

= 0 implies full surface patch test plate modes.

= 1 implies filamentary test plate modes.

2. READ's 2-5

READ's 2-5 specify the far-zone pattern calculations desired. READ's 2 and 3 specify the elevation and azimuth radiation patterns, respectively. READ's 4 and 5 specify the elevation and azimuth scattering patterns, respectively.

READ 2 defines the following:

IFE = indicator for calculating the far zone elevation radiation pattern.

= 0 implies do not compute far zone radiation pattern in the elevation plane.

= 1 implies compute far zone radiation pattern in the elevation plane.

IPFE = plot indicator.

= 0 implies do not plot far zone radiation pattern in the elevation plane.

= 1 implies plot far zone radiation pattern in the elevation plane.

NDFE = angle increment in degrees for far zone radiation pattern in the elevation plane.

PHFE = phi angle in degrees for far zone radiation pattern in the elevation plane.

READ 3 defines the following:

IFA, PFA, NDFA = same as IFE, IPFE, NDFE but for azimuth plane.

THFA = theta angle in degrees for far zone radiation pattern in the azimuth plane.

READ 4 defines the following:

ISE = indicator for calculating the far zone elevation plane scattering pattern. Scattering implies either backscattering (ISE = 1) or bistatic scattering (ISE = 2).

= 0 implies do not compute far zone scattering pattern in the elevation plane.

= 1 implies compute backscattering pattern in the elevation plane.

= 2 implies compute bistatic scattering in the elevation plane.

IPSE = plot indicator.

= 0 implies do not plot far zone scattering pattern in the elevation plane.

= 1 implies plot far zone scattering pattern in the elevation plane.

NDSE = angle increment in degrees for far zone scattering pattern in the elevation plane.

PHSE = phi angle in degrees for far zone scattering pattern in the elevation plane.

THIN = theta angle of the incident wave for bistatic scattering calculations (i.e., ISE = 2 or ISA = 2).

PHIN = phi angle of the incident wave for bistatic scattering calculations.

READ 5 defines the following:

ISA,IPSA,NDSA = same as ISE,IPSE,NDSE but for scattering in the azimuth plane.

THSA = theta angle in degrees for far zone scattering pattern in the azimuth plane.

NOTES:

If ISA or ISE are set to -1 or -2, then the image of the incident wave is included for the azimuth or elevation scattering calculations, respectively. This option is useful for treating problems over an infinite ground plane using image theory. The image of the geometry of the scatterer structure has to be defined in the input file. However, the program defines the image incident wave automatically.

On the same run one can obtain one of each, i.e., either a radiation pattern or a scattering pattern or a bistatic scattering pattern. For each type of calculation one can obtain both polarizations. For more information see the output section of SUBROUTINE SORTB in Chapter IV.

To obtain different patterns of the same antenna or scatterer structure see READ 10 input statement.

3. READ 6

READ 6 defines the following:

FMC = frequency in megahertz.

CMM = wire conductivity in megamohms/meter. CMM = -1.0 implies a perfect conductor.

A = the wire radius in meters.

4. READ'S 7-9

READ's 7-9 define the plate geometry. In particular READ 7 defines the following:

NPLTS = the total number of plates.

READ 8 defines the following for every plate NPL:

NCNRS(NPL) = the number of corners on plate NPL.

SEGM(NPL) = the maximum segment size of the surface patch monopoles on plate NPL (in wavelengths). It should not exceed 0.25λ and is typically chosen 0.25λ. If more accuracy is needed SEGM can be chosen less than 0.25λ with a substantial sacrifice of computation time since the number of modes increases.

IREC(NPL) = rectangular/polygonal plate indicator for plate NPL.

= 0 implies plate NPL is polygonal.

= 1 implies plate NPL is rectangular.

IPN(NPL) = polarization indicator. It has meaning only for quadrilateral plates.

= 1 implies modes are to be placed on the quadrilateral plate NPL to cover the first polarization only.

= 2 implies modes are to be placed on the quadrilateral plate NPL to cover the second polarization only.

= 3 implies both polarizations are to be placed on plate NPL. Also for a polygonal plate NPL IPN(NPL) = 3.

The term first polarization implies the current flowing in the direction of side 1-2. The second polarization implies the current flowing in the direction of side 2-3.

IGS(NPL) = number of generating side in SUBROUTINE PLATE3. The generating side is the reference side subroutine PLATE3 uses to divide plate NPL into modes (see description of PLATE3 in chapter 4). Normally IGS(NPL) = 0 which implies that subroutine PLATE3 will use the longest side of the plate NPL as the generating side. However, the ensuing modal segmentation may not be the optimal one in the sense of minimum number of modes and accurate representation of the current flow on the plate. In such cases the user might want to use a different generating side by setting IGS(NPL) = the side number of the desired reference side, i.e., 0 < IGS(NPL) < NCNRS(NPL).

The program automatically checks for plates which intersect along a common edge and inserts surface patch overlap modes to ensure the continuity of current along the common edge. If more than two plates interesect along a common edge the program finds the minimum linearly independent set of overlapping plates. For a detailed description of how overlap plate modes are defined see SUBROUTINE POPLOV, Chapter IV. READ 9 statement inputs the coordinates of the corners of plate NPL. It is executed NCNRS(NPL) times for plate NPL and it defines the following:

PCN(1,NCNR,NPL),(PCN(2,NCNR,NPL),PCN(3,NCNR,NPL) = x,y,z coordinates ,respectively, of the corner NCNR (1 < NCNR < NCNRS(NPL)) of plate NPL (in meters).

5. READ 10

At times a user may wish to run several consecutive problems for which the impedance matrix either does not change or only certain blocks of it change. For example, the impedance matrix will not change for the following cases:

1. if different far-zone patterns are desired,
2. if different voltage excitations are used, or

3. if different angles of incidence are used in a bistatic scattering calculation.

Obviously in these cases it would be extremely wasteful to recompute the entire impedance matrix. At other times the geometry may change only slightly from one run to the other. For example, consider the problem of locating a monopole on a ship such that a desired impedance and/or pattern is achieved. In order to solve this problem one would construct a model of the ship from several intersecting plates, possibly requiring hundreds of surface patch modes. One attachment mode would be required where the monopole physically connects to a plate. The user would then analyze this configuration for many monopole locations in search of the optimum location. The impedance matrix of this (and in general of any) MM problem can be visualized as shown in Figure 3-2 . As the monopole is moved around, the P/P block of the matrix does not change; only the blocks involving wire and attachments change. Thus a considerable savings in time will result if on the first run the entire matrix is stored on a disk file. On subsequent runs the stored matrix is read in and only the blocks involving wires and attachments are recomputed.

The operation of storing, reading and selecting the blocks of the impedance matrix to be recomputed is controlled by the parameters IWRZM and IRDZM. Specifically:

IWRZM = indicator for writing the impedance matrix on
a disk file.
= 0 implies do not write out the impedance matrix.

$$\left[\begin{array}{c|c|c} {}^{W}/_{W} & {}^{W}/_{P} & {}^{W}/_{A} \\ \hline {}^{P}/_{W} & {}^{P}/_{P} & {}^{P}/_{A} \\ \hline {}^{A}/_{W} & {}^{A}/_{P} & {}^{A}/_{A} \end{array}\right]$$

W = WIRE
P = PLATE
A = ATTACHMENT

Figure 3-2. Symbolic representation of the nine different blocks of the impedance matrix.

= 1 implies write out the impedance matrix.

IRDZM = indicator for reading in the impedance matrix calculated during the previous run.

= 0 implies do not read in the previous matrix and calculate the entire new matrix.

= 1 implies read in the previous matrix and compute new matrix except for the W/W and A/A blocks.

= 2 implies read in the previous matrix and compute new matrix except for the P/P block.

= 3 implies read in old matrix and use as new matrix, i.e., do not calculate any impedance elements.

NOTES:

Thus IRDZM = 2 if the plate geometry is unchanged from the last run, IRDZM = 1 if the wire and attachment geometry is unchanged from the last run and IRDZM = 3 if the entire geometry is unchanged.

Whenever IRDZM>0 the following should be true:

1. IWRZM must be 1 on the previous or first run, and
2. the number of wire, plate and wire attachment modes is unchanged from the IWRZM = 1 run.

The impedance matrix is read from and is written on the disk file ZMAT.DAT on logical unit 1.

6. READ'S 11-15

The following read statements input the wire geometry of the problem including the impedance loads, voltage generators and wire-to-plate attachments. These read statements are executed only if INWR = 1 and IRGM = 1 (see READ 1). The wire geometry input will be described with the aid of the example shown in Figure 3-3. The structure consists of a T-shaped wire with one load and one generator. The wire is defined by four points, shown as heavy black dots in Figure 3-3, and three wire segments. The wire point numbering scheme shown in Figure 3-3 is arbitrary. The wire point numbers are shown adjacent to the dots and the segment numbers are shown as circled numbers next to the segments.

The following rules apply for wires:

1. The wire geometry consists of interconnected straight wire segments.
2. Each segment should not exceed a quarter wavelength in length.
3. Two intersecting segments should form an acute angle no less than 30 degrees.
4. Single isolated wire segments are not permitted.
5. There is no limit to the number of wire segments intersecting at a given point.

READ 11 inputs the following parameters:

NM = total number of segments on the wire structure.

NP = total number of points on the wire structure.

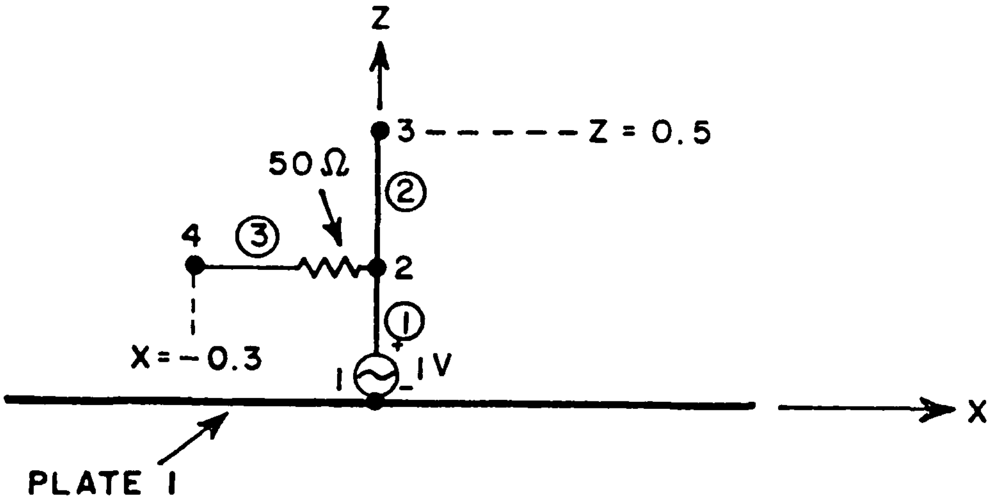


Figure 3-3. A wire geometry showing points, segments, a load, a generator and an attachment point.

NAT = total number of wire-to-plate attachment points.

NFPT = total number of feed locations on the wire structure, excluding feeds at wire-to-plate attachment points.

There are times when the mutual coupling between two feed locations (points) on the wire structure is needed. By specifying

NFS1 = wire "location" of the first feed point, and

NFS2 = wire "location" of the second feed point,

the program will calculate the maximum coupling between feed points NFS1 and NFS2. Also, the impedance matrix relating the two feed points is calculated. If no coupling calculations are needed then NFS1 = NFS2 = 0.

The term wire "location" implies either endpoint A or endpoint B of a particular segment. Specifically, if endpoint A of segment L is meant then the wire "location" is L. If endpoint B is meant then the wire "location" is L + NM, where NM is the total number of wire points.

For the example of Figure 3-3 READ 11 would be:

3 4 1 1 0 0 (no coupling is specified).

READ 12 requires NP lines of inputs to define the x,y,z coordinates of every point on the wire structure (in meters):

X(I) = the x coordinate of point I.

Y(I) = the y coordinate of point I.

Z(I) = the z coordinate of point I.

For the geometry of Figure 3-3 the NP = 4 lines of input for READ 12 are:

```
 0.0 0.0 0.0
 0.0 0.0 0.25
 0.0 0.0 0.5
-0.3 0.0 0.25   .
```

READ 13 requires NM lines of input to define the endpoints of every segment. Each segment has two endpoints denoted by A and B. The user can arbitrarily select which end is A and which is B. READ 13 defines:

IA(J) = endpoint A of wire segment J.

IB(J) = endpoint B of wire segment J.

By arbitrarily choosing the endpoint with the smaller point number as A, the NM = 3 lines of input for READ 13 would be:

```
1 2
2 3
2 4    .
```

READ 14 defines for every feed point its wire "location" and the complex value of the generator and load at that location. In this code one always think of genenators and loads as being inserted into segments, either by endpoint A or B of the segment. One should not think of feeds as being by a point in the wire. For example, for the geometry of Figure 3-3 it is not sufficient to specify a 50 ohm load by point 2. There are three locations (although physically close, electrically very different) which could be taken as point 2, i.e., endpoint B of segment 1, endpoint A of segment 2, or endpoint A of segment 3. The last location is the correct location specification for the 50 ohm load. READ 14 defines the following:

IFM = segment number of feed point.

IAB = indicator specifying by which endpoint of segment IFM the feed point is located.

= 0 implies feed point is by endpoint A of segment IFM.

= 1 implies feed point is by endpoint B of segment IFM.

VLG = complex voltage generator at the feed point (in volts). Positive polarity is from endpoint A to endpoint B of segment IFM.

ZL = complex impedance loading at the feed point (ohms).

For the geometry of Figure 3-3 the NFPT = 1 line of input for READ 14 would be:

```
3 0 (0.0,0.0) (50.0,0.0)  .
```

Note that there is no voltage generator at this wire "location".

READ 15 specifies the wire-to-plate attachment geometry along with the complex values of the generators and loads at the attachment locations. Specifically, READ 15 defines the following for each of the NAT attachments:

NAS = the number of the segment which attaches to the plate.

IAB = indicator specifying which endpoint of segment NAS attaches to the plate.

= 0 implies endpoint A of segment NAS.

= 1 implies endpoint B of segment NAS.

NPLA(I) = plate where the attachment point I is located.

VGA(I) = complex voltage generator at attachment point I.

ZLDA(I) = complex impedance loading at the attachment point I.

BDSK(I) = outer disk radius in meters of the disk monopole of the Ith attachment mode. Experience has shown that for accurate impedance and pattern results the disk radius should be between 0.1λ and 0.25λ. A good average choice is 0.2λ. Also, the center of the disk should be at least 0.1λ away from all plate edges.

Assuming a frequency of 300.0 Mhz, READ 15 for the geometry of Figure 3-3 would require NAT = 1 lines of input:

```
     1 0 1 (1.0,0.0) (0.0,0.0) 0.2  .
```

Note that there is no impedance load at the the attachment point.

B. SUBROUTINE WGEOM

If INWR = 1 and IRGM = 0 (See READ 1), then the wire and attachment geometry is defined by subroutine WGEOM, which has to be written by the user. The general form of subroutine WGEOM is:

```
SUBROUTINE WGEOM(IA,IB,X,Y,Z,NM,NP,NAT,NSA,NPLA,VGA,BDSK,
ZLDA,NWG,VG,ZLD,WV,NFS1,NFS2)

DIMENSION IA(1),IB(1),X(1),Y(1),Z(1),NSA(1),NPLA(1),BDSK(1)
COMPLEX VGA(1),ZLDA(1),VG(1),ZLD(1)

             .          .
             .          .

MAIN PROGRAM
RETURN
END
```

The following parameters are inputs and are defined in the main program:

NWG = indicator for the number of wire geometries.

WV = wavelength.

The following parameters are outputs:

IA(I) = endpoint A of wire segment I (I = 1,NM).

IB(I) = endpoint B of wire segment I (I = 1,NM).

X(J),Y(J),Z(J) = x,y,z coordinates of point J (J =1,NP).

NM = the total number of wire segments.

NP = the total number of wire points.

NAT = the total number of attachment points.

NSA(K) = wire "location" of attachment point K.

NPLA(K) = the number of the plate where attachment point K is located.

VGA(K) = complex voltage generator at attachment point K.

BDSK(K) = outer disk radius of the disk monopole of attachment mode K.

ZLDA(K) = complex impedance load at attachment point K.

VG(L) = complex voltage generator at wire
"location" L.

ZLD(L) = complex impedance load at wire
"location" L.

NFS1 = wire "location" of the first feed point.

NFS2 = wire "location" of the second feed point.

The parameters NFS1 and NFS2 are used when the mutual coupling between two feed points on the wire structure is required. When no mutual port coupling calculation is needed then NFS1 = NFS2 = 0.

All of the above outputs must be defined by the user via FORTRAN statements in subroutine WGEOM. Usually WGEOM is written on a seperate file and is linked with the main program. This procedure saves compiling time when debugging or changing WGEOM.

1. Example of WGEOM Subroutines:

Consider the problem of the center-fed dipole. If one wants to study different dipole lengths and/or segmentations, it is more efficient to write a subroutine to generate the dipole geometry for arbitrary length and segmentation. An arbitrary dipole consisting of NM segments and DH segment length is shown in Figure 3-4. A subroutine describing the geometry of this arbitrary dipole should define the following parameters:

1. The number of points NP (NP = NM + 1).
2. The segment size DH (DH = H/NM).

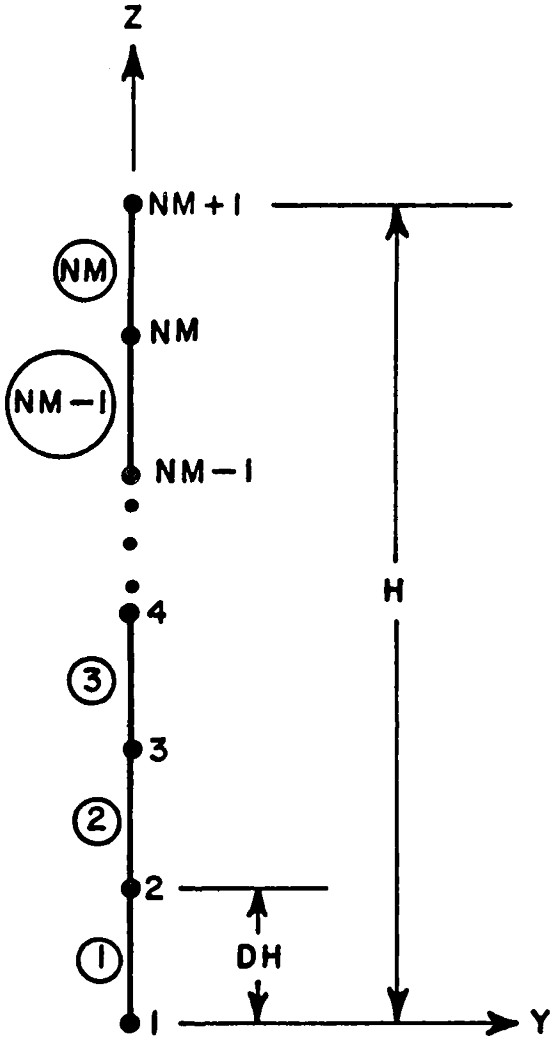


Figure 3-4. Segmentation of a straight wire.

3. The coordinates X(I),Y(I),Z(I) of the Ith point, i.e.,

   X(I) = 0.0

   Y(I) = 0.0

   Z(I) =DH*(I-1) .

4. The endpoints A and B of segment J, i.e.,

   IA(J) = J

   IB(J) = J + 1 .

5. If the dipole is to be center fed then NM must be an even number and the generator wire "location" is:

   IGN = (NM/2) + 1 or

   IGN = NM + NM/2 .

6. There are no attachments, i.e., NAT=0.

7. No coupling calculations are desired, i.e., NFS1=0, NFS2=0.

A possible WGEOM subroutine to handle the center-fed dipole is shown in Figure 3-5. The length of the dipole is set at H = 0.5 meters and the number of segments NM = 4. The advantage of writing a subroutine WGEOM for this problem is that a user can obtain dipoles of different lengths and/or segmentations. This can implemented by simply changing the parameters H amd NM. Otherwise, for every different dipole length or segmentation a whole new wire geometry would have to be defined.

As a second example consider the problem of describing a regular polygon loop of arbitrary radius, number of sides and segment size. Let:

```
      SUBROUTINE WGEOM(IA,IB,X,Y,Z,NM,NP,NAT,NSA,NPLA,VGA,BDSK,
     2 ZLDA,NWG,VG,ZLD,WV,NFS1,NFS2)
      DIMENSION IA(1),IB(1),X(1),Y(1),Z(1),NSA(1),NPLA(1),BDSK(1)
      COMPLEX VGA(1),ZLDA(1),VG(1),ZLD(1)
C
C     GEOMETRY FOR A CENTER FED DIPOLE
C
C     SPECIFY H = WIRE LENGTH AND NM = NUMBER OF SEGMENTS
      H=0.5
      NM=4
C     INSURE THAT NM IS EVEN
      NM=2*((NM+1)/2)
C     THE NUMBER OF POINTS IS
      NP=NM+1
C     THE SEGMENT SIZE IS
      DH=H/NM
C     DEFINE COORDINATES OF NP POINTS AND NM SEGMENTS
      DO 100 I=1,NP
      X(I)=0.0
      Y(I)=0.0
      Z(I)=(I-1)*DH
      IA(I)=I
      IB(I)=I+1
100   CONTINUE
C     DEFINE GENERATOR LOCATION AND VALUE
      IGN=NM/2+1
      VG(IGN)=(1.0,0.0)
C     INDICATE NO ATTACHMENTS
      NAT=0
C     INDICATE NO COUPLING
      NFS1=0
      NFS2=0
      RETURN
      END
```

Figure 3-5. A subroutine WGEOM to describe the center fed dipole of Figure 3-4.

R = the loop radius in meters.

NS = the number of sides on the loop.

SWX = the maximum segment size in λ.

Figure 3-6 shows a hexagon loop with two segments per side. For a general loop subroutine, WGEOM should define the following parameters:

1. The length SL of each side.
2. The number of segments per side (NMS) and the length of each segment (DSL).
3. The total number of segments (NM = NMS*NS) and the total number of points (NP = NM).
4. The x,y,z coordinates of the endpoints of side I, i.e.,

   X1 = R*COS(PH1), PH1 = (I-1)*360/NS

   Y1 = R*SIN(PH1)

   X2 = R*COS(PH2), PH2 = I*360/NS

   Y2 = R*SIN(PH2) .
5. The x,y,z coordinates of point K which is point J on side I, i.e.,

   K = (I-1)*NMS + J

   X(K) = X1 + (J-1)*DX12, DX12 = (X2 - X1)/NMS

   Y(K) = Y2 + (J-1)*DY12, DY12 = (Y2 -Y1)/NMS

   Z(K) = 0.0 .
6. The endpoints A and B of segment K are:

   IA(K) = K

   IB(K) = K + 1 except IB(NM) = 1 .

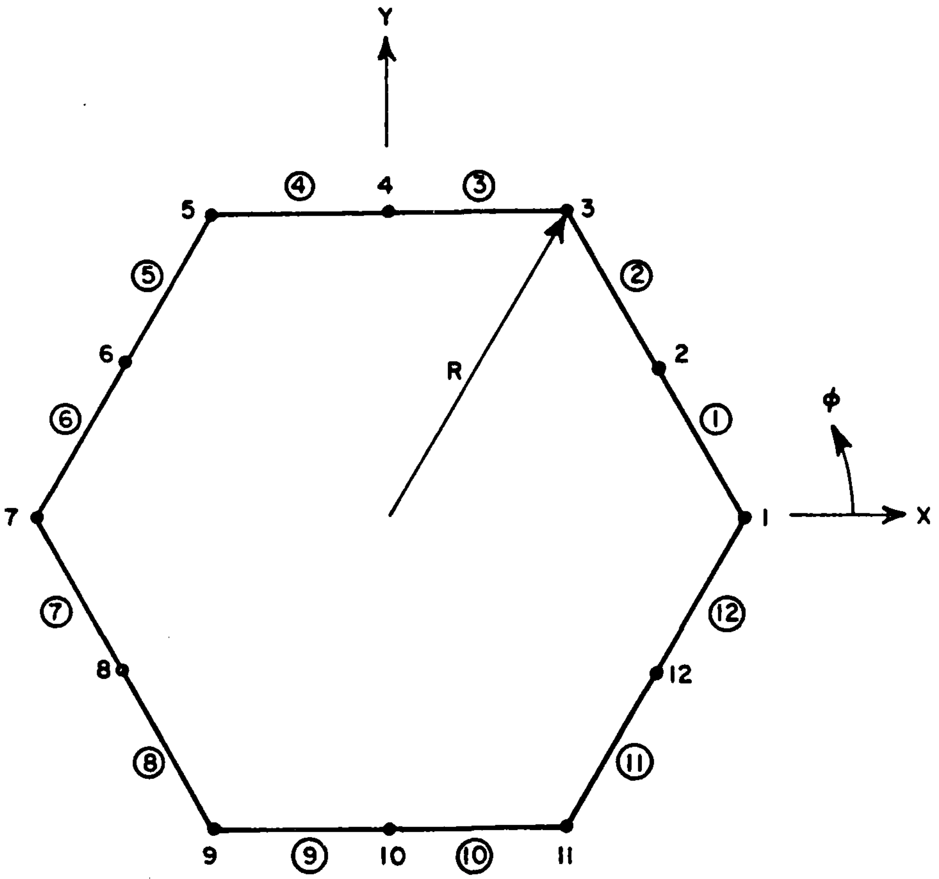


Figure 3-6. Segmentation of a hexagon loop.

7. The feed point is by endpoint A of segment 1, i.e., IGN = 1.

8. There are no attachments, i.e., NAT=0.

9. No coupling calculations are desired, i.e., NFS1=0 NFS2=0.

A possible WGEOM subroutine is shown in Figure 3-7. Here NS = 6, R = 0.3 m and SWX = 0.2λ. The advantage of writing a subroutine WGEOM for this problem is that a user can define regular polygon loops with different radii, number of sides and segment sizes. This can be accomplished by changing only the parameters R, NS and SWX. Also, since SWX is specified in wavelengths the subroutine is frequency independent. This feature is especially desirable for an analysis of the wire antenna over a broad frequency range.

C. ARRAY DIMENSIONS

The array dimensions are defined by DIMENSION and COMPLEX statements at the top of the main program. All arrays have either fixed dimensions, independent of the geometry being run, or are dimensioned according to one of the following dimension indicators:

INM = maximum number of wire segments.

ICJ = maximum number of wire modes.

IPLM = maximum number of plate modes.

IPL = maximum number of plates.

IAT = maximum number of wire to plate attachments.

INP = maximum number of wire points.

```
      SUBROUTINE WGEOM(IA,IB,NM,NP,NAT,NSA,NPLA,VGA,BDSK,
     2 ZLDA,NWG,VG,ZLD,WV,NFS1,NFS2)
      DIMENSION IA(1),IB(1),X(1),Y(1),Z(1),NSA(1),NPLA(1),BDSK(1)
      COMPLEX VGA(1),ZLDA(1),VG(1),ZLD(1)
C
C     GEOMETRY FOR POLYGONAL LOOP
C
C     SPECIFY R= LOOP RADIUS IN METERS, NS= NUMBER OF
C     SIDES IN POLYGONAL LOOP, AND SWX=MAXIMUM SEGMENT SIZE
C     IN WAVELENGTHS.
      R=0.3
      NS=6
      SWX=0.2
C     FIND SL= SIDE LENGTH
      PI=4.0*ATAN(1.0)
      DPH=2.0*PI/NS
      SL=2.0*R*SIN(DPH/2.0)
C     FIND NMS= NUMBER OF SEGMENTS PER SIDE AND DSL= THE
C     SEGMENT LENGTH.
      DSL=SWX*WV
      DSL=SL/NMS
      NMS=0.99 + SL/DSL
C     FIND NM= THE TOTAL NUMBER OF SEGMENTS AND NP= THE TOTAL
C     NUMBER OF POINTS.
      NM=NS*NMS
      NP=NM
C     DEFINE NMS POINTS AND SEGMENTS ON EACH OF THE NS SIDES
      DO 100 I=1,NS
C     THE COORDINATES OF THE FIRST END OF SIDE I
      PH1=(I-1)*DPH
      X1=R*COS(PH1)
      Y1=R*SIN(PH1)
C     THE COORDINATES OF THE SECOND END OF SIDE I
      PH2=I*DPH
      X2=R*COS(PH2)
      Y2=R*SIN(PH2)
C     EACH POINT ON SIDE I WILL BE
      DX12=(X2-X1)/NMS
      DY12=(Y2-Y1)/NMS
C     FROM THE LAST POINT ON SIDE I
      DO 200 J=1,NMS
C     DEFINE THE KTH POINT AND SEGMENT
      K=(I-1)*NMS + J
      X(K)=X1+(J-1)*DX12
      Y(K)=Y1+(J-1)*DY12
      Z(K)=0.0
      IA(K)=K
      IB(K)=K+1
      IF(K.EQ.NM)IB(K)=1
200   CONTINUE
100   CONTINUE
C     PLACE A 1 VOLT GENERATOR AT THE X AXIS
      IGN=1
      VG(IGN)=(1.0,0.0)
C     INDICATE NO ATTACHMENTS
      NAT=0
C     INDICATE NO COUPLING
      NFS1=0
      NFS2=0
      RETURN
      END
```

Figure 3-7. A subroutine WGEOM to describe the hexagon loop of Figure 3-6.

ITOT = maximum total number of modes (wire + plate + attachment) if full surface patch test plate modes are used (IFIL = 0).

IDZT = maximum number of entries in impedance array ZT(IDZT).

ICC = maximum number of modes if filamentary test plate modes are used (IFIL = 1). ICC is the dimension indicator for impedance matrix ZTF(ICC,ICC).

The dimensions indicators are defined below the DIMENSION and COMPLEX statements and typically have the values:

INM = 491

ICJ = 492

IPLM = 490

IPL = 30

IAT = 2

INP = 493

ITOT = 495

IDZT = (ITOT*ITOT + ITOT)/2

ICC = 360 .

Because of limited memory allocation space, when IFIL = 0 then ICC should be set to 1. Similarly, when IFIL = 1, IDZT should be set to 1. Also note that while the number of wire modes can be up to 492, the number of plate modes up to 490 or the number of attachment modes up to

4, the total number of modes can not exceed 495, if IFIL = 0, or 360 if IFIL = 1.

Two steps are required in order to change the dimensions:

1. Change the appropriate dimension indicator.
2. Re-dimension all the arrays associated with that dimension indicator.

Arrays dimensioned by the same indicator are grouped together and are clearly identified by COMMENT statements at the beginning of the main program.

D. PROGRAM FILE DESCRIPTIONS

The computer code is contained in several files stored on disk in the ElectroScience Laboratory's computer, which is a VAX 11/780 manufactured by DIGITAL EQUIPMENT CORPORATION. A listing of the FORTRAN version of the files follows (except for 'PLOTLIB which is an object file):

STDMM2.FOR - the main program plus all the subroutines except the thin wire subroutines.

THNWRS.FOR - thin wire subroutines.

WGEOM.FOR - subroutine describing the wire structure geometry, written by the user (see section 2.3).

'GRP11LIB - contains various special library subroutines. At present only the function subroutine GETCP(I) is used where I = clock reading in hundreths of a second. Since this subroutine tends to be hardware dependent, it is not included when the program is sent outside the ElectroScience Laboratory.

GPLOT2.FOR - subroutine for making three-view orthographic plot of wire and plate geometry.

'PLOTLIB - contains various plotting subroutines. When the program is sent outside the ElectroScience Laboratory some routines must be omitted due to contactual restrictions. When this file is supplied to an outside user it will be called PLOTLIB.FOR. Of the subroutines omitted, the only four used in the program are:

VPLOTS(I,0,0) - reserves the plotter.

I = 1 implies the plot is for the Versatek paper plotter.
= 2 implies the plot is for the Megatek plotter.
= 0 implies the program gives the user a choice of plotter.

PLOT(X,Y,I) - moves the plotter "pen", with pen up or down.

X,Y = the x,y coordinates of the point where the pen is going.

I = 2 implies the pen is lowered before moving.
= 3 implies the pen is raised before moving.
= -2 or -3 implies the same as 2 or 3 except that the origin is reset after moving.
= -999 implies go to the lower left corner of next page with pen up and reset the origin.

= 999 implies this is the last plotting call, i.e., all plotting is terminated by calling PLOT(X,Y,999).

NUMBER(X,Y,HT,FPN,ANGLE,N) - plots out a floating point number.

X,Y = are the x,y coordinates of the lower corner of the number in inches.

HT = height of the output number in inches. If HT>0, then the output is plotted to the right of X,Y; if HT<0, it will be plotted to the left of X,Y.

FPN = floating point number to be plotted.

ANGLE = angle in degrees(counter clockwise) with respect to the X axis at which FPN is to be plotted.

N = integer specifying the output format. If the absolute value of N is less than 100, then F.N will be plotted in the "F" format. If N>0 then N digits will be plotted after the decimal point, in addition to all the digits before the decimal point. If N<0, then no digits will be plotted after the decimal point and the decimal point plus the first -(N + 1) digits to the left of the decimal point will be supressed. If the absolute value of N is larger than 100, then FPN will be plotted in the "E" format or expontential scientific format. If N>100, there will be one digit to the left and N - 100 digits to the right of the decimal point in the mantissa. If N<-100, then the mantissa will be an integer with -(N + 100) digits.

SYMBOL(X,Y,HT,LABEL,ANGLE,NC) - plots a character or string of characters.

X,Y = coordinates in inches of the lower left hand corner of the symbol to be drawn.

HT = the height in inches of the character to be drawn. HT should be a multiple of 7 times the plotter increment.

LABEL = if NC>0 LABEL is a literal variable or constant representing the character strung to be plotted. NC is the number of characters to be plotted. If NC = -1, then LABEL is an integer expression, ranging from 0 to 127, which represents a single character. These symbol and their codes are shown in Appendix 40.

ANGLE = angle in degrees between the symbol to be plotted and the X axis.

NC = see description of LABEL.

IF the user cannot supply a subroutine GETCP, then all calls to this subroutine should be deleted and the program will not supply the run time information at the end of the program run. If subroutines VPLOTS, PLOT, NUMBER and SYMBOL are not available in the system, all calls to them should be commented out. If plotting is not desired, then the calls to subroutine GPLOT2, MPLOT, and MPLOT2 should be removed along with all calls to VPLOTS, PLOT, NUMBER and SYMBOL.

In summary, when the code is supplied outside the ElectroScience Laboratory the following FORTRAN files are included in a single file called OSUESP.FOR:

STDMM2.FOR
THNWRS.FOR
WGEOM.FOR
GPLOT2.FOR
PLOTLIB.FOR .

Note that the WGEOM.FOR supplied is for a dipole antenna. To obtain a new geometry the user must write a new subroutine WGEOM and replace the one supplied.

E. DESIGN EXAMPLES

This section will present several design or example runs illustrating the use of the code. The purpose of the examples is:

1. to illustrate the input data,
2. to illustrate the output data, and
3. to provide trial or debugging runs for a new user.

1. DESIGN EXAMPLE 1

The currents, input impedance and far-zone elevation plane radiation patterns in the phi = 0.0 plane for the geometry of Figure 3-3 are desired. The wire is located in the center of a one meter square plate. The frequency is 150 MHz, the wire is made of aluminum (conductivity = 38 megamhos/meter) and the wire radius is 0.001 meter. The input file for this problem is shown below:

```
READ 1-----1 2 1 1 1 0 4 10 18 1 1 1
READ 2-----1 1 3 0.0
READ 3-----0 1 3 90.0
READ 4-----0 1 3 0.0 90.0 0.0
READ 5-----0 1 3 90.0
READ 6-----150.0 38.0 0.001
READ 7-----1
READ 8-----4 0.25 1 3 0
READ 9-----0.5 -0.5 0.0
           0.5 -0.5 0.0
           0.5 0.5 0.0
           -0.5 0.5 0.0
READ 10----1 0
READ 11----3 4 1 1 0 0
READ 12----0.0 0.0 0.0
           0.0 0.0 0.25
           0.0 0.0 0.5
           -0.3 0.0 0.25
```

```
READ 13----1 2
           2 3
           2 4
READ 14----3 0 (0.0,0.0) (50.0,0.0)
READ 15----1 0 1 (1.0,0.0) (0.0,0.0) 0.4 .
```

Note that filamentary test plate modes are used, i.e., IFIl = 1 in READ 1.

Before computing any patterns or data the accuracy of the geometry as defined by the input file should be checked. A three-view orthographic plot of the geometry , shown in Figure 3-8, is obtained by setting NGO = 0. The edges of the plate are shown as solid lines. Wire segments are shown as solid lines with small circles representing the endpoints. This plot can be used as a first check of the accuracy of the input file.

The output for this problem is shown in Appendix 1 and could be broken in the following blocks:

a. Input Data

A listing of some of the input quantities such as frequency, wire radius, wire conductivity, integration parameters and the indicator for the type of test plate modes.

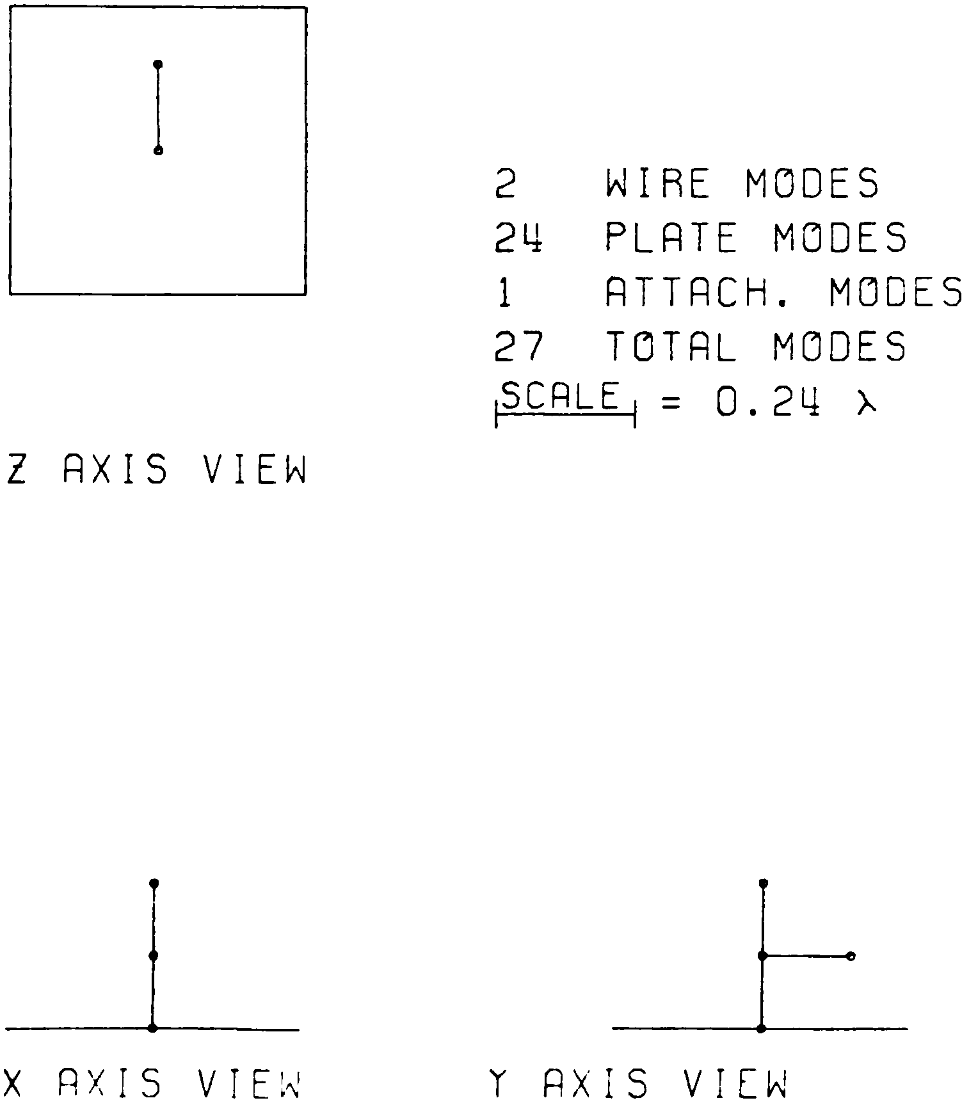


Figure 3-8. Three-view plot of the geometry of Example 1.

b. Plate Geometry

For every plate its type is specified (rectangular or polygonal) along with the coordinates of every plate corner, the maximum segment size (SEGM), the polarization indicator (IPN) and the generating side indicator (IGS). Since IWR = 1 a detailed printout of the x,y,z coordinates of every surface patch monopole on the plate is included. Figure 3-9 shows a typical surface patch dipole mode consisting of monopole A and monopole B. Monopole A is defined by the x,y,z coordinates of its four corners A1,A2,A3,A4, while monopole B is defined by its four corners B1,B2,B3,B4. By convention positive current flows from monopole A to monopole B.

c. Wire Geometry

First the x,y,z coordinates of the NP wire points are printed along with the the maximum and minimum number of modes at any point. Since IWR = 1, the wire modal layout is printed by specifying I1, I2, I3, JA, JB for every wire mode. Figure 3-10 shows a typical wire dipole mode defined by points I1, I2,I3 and segments JA and JB. By convention positive current flows from JA to JB. Next the endpoints and length of the NM wire segments are printed.

d. Attachment Geometry

For every attachment point the following parameters are printed:

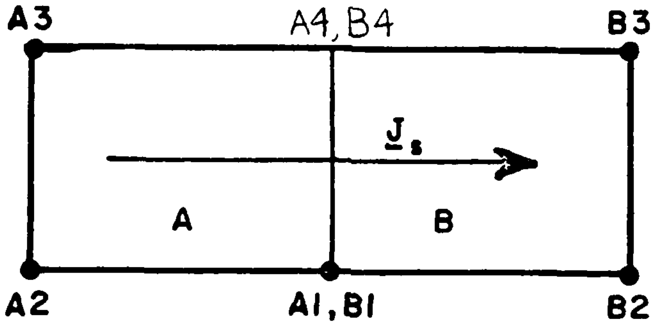


Figure 3-9. A surface patch dipole mode.

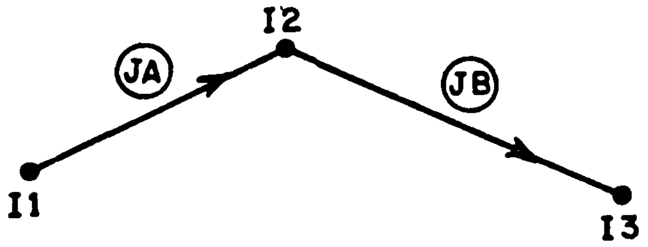


Figure 3-10. A surface patch dipole mode.

SEGMENT = the wire segment which attaches to the plate.

END = same as IAB, the indicator for which wire segment endpoint attaches to the plate.

PLATE = the plate where the wire segment attaches to.

B = outer disk radius of the disk monopole of the attachment mode.

e. Loads And Generators

The wire "location" and complex value of every impedance load and voltage generator is printed. Finally the total number of wire, plate and attachment modes is printed. The output file ends here if NGO = 0. The user should check the output file carefully to make sure everything is defined correctly.

If NGO = 1 the program proceeds with the calculations and outputs the following additional information:

f. Antenna Modal Currents

If IWR = 1, the induced modal currents are printed. For every modal current its relative magnitude (with respect to the largest modal current), its absolute magnitude (in amperes) and its phase (in degrees) are printed.

g. Antenna Impedance And Patterns

The input impedance shown is for a unit voltage generator, i.e., (1 + j0) volts. The far-zone patterns are printed as:

GTHETA = the gain in db of the θ component of the electric field.

GPHI = the gain in db of the ϕ component of the electric field.

Figures 3-11a and b are the polar plots of GTHETA and GPHI, respectively. At the end of the file the total CPU time for the run is printed.

2. DESIGN EXAMPLE 2

The backscattering from the corner reflector of Figure 3-12 is desired. It consists of two 1.0λ by 0.5λ rectangular plates intesecting along the z-axis. The pattern is to be taken in the azimuth plane at theta = 90.0 degrees. This is specified in the input file by setting ISA = 1 and THSA = 90.0. The input file for this problem is as follows:

```
READ 1----1 2 1 1 0 0 4 10 18 0 1 1
READ 2----0 1 3 0.0
READ 3----0 1 3 90.0
READ 4----0 1 3 90.0 90.0 45.0
READ 5----2 1 3 90.0
READ 6----300.0 -1.0 0.001
```

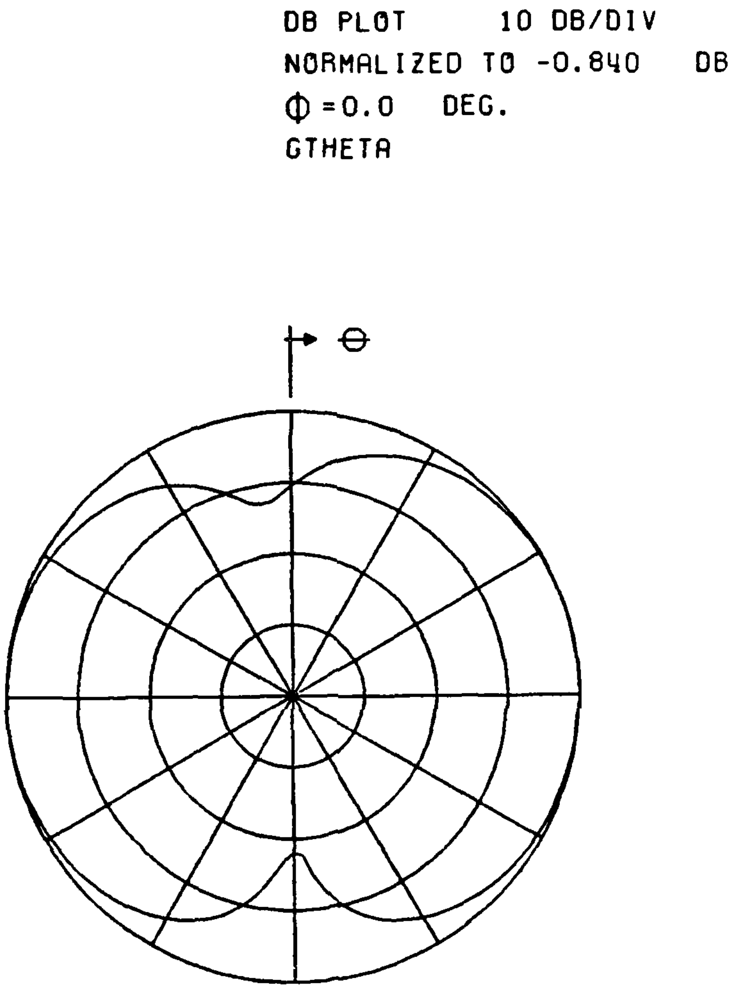


Figure 3-11. (a) θ polarized radiation pattern, and

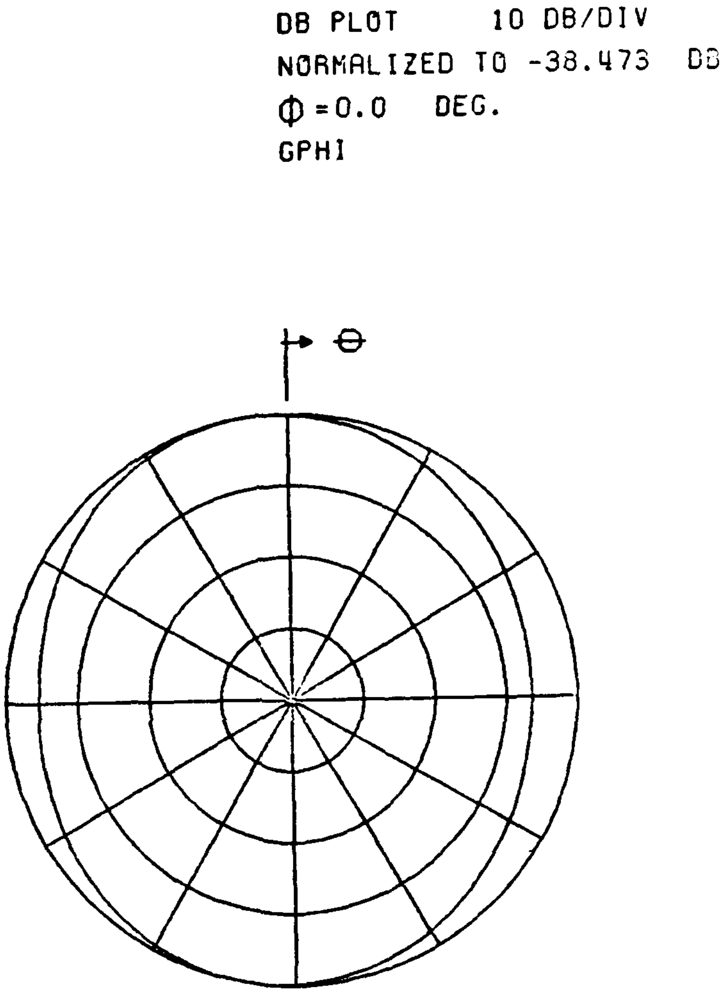


Figure 3-11. (b) ϕ polarized radiation pattern.

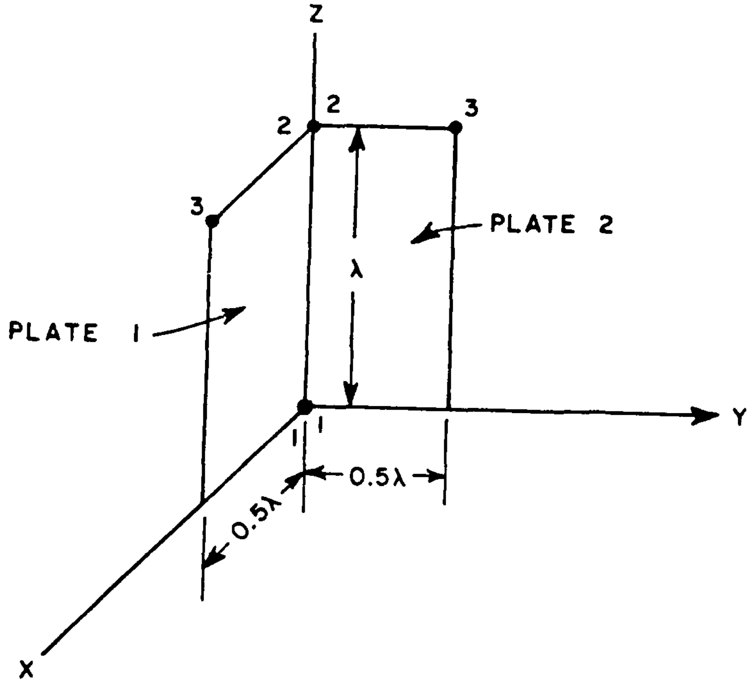


Figure 3-12. Geometry for the corner reflector of Example 2.

```
READ 7----2

READ 8----4 0.25 1 3 0

READ 9----0.0 0.0 0.0

          0.0 0.0 1.0

          0.5 0.0 1.0

          0.5 0.0 0.0

READ 8----4 0.25 1 3 0

READ 9----0.0 0.0 0.0

          0.0 0.0 1.0

          0.0 0.5 1.0

          0.0 0.5 0.0

READ 10---1 0.
```

Figure 3-13 shows the three-view orthographic plot, obtained by setting NGO = 0. Also, if IWR = 1, plots of the modal layouts on both plates are obtained (Figures 3-14a and b) along with a plot of the overlap modal layout (Figure 3-14c). The output for this problem is given in Appendix 2. The program automatically inserts the necessary overlap modes between the two plates. Note that after specifying the x,y,z coordinates of the corners of the two plates, the output indicates that four overlap modes were inserted between side 1 of plate 1 and side 1 of plate 2.

Finally, a printout of all the various cross sections is included, i.e.,

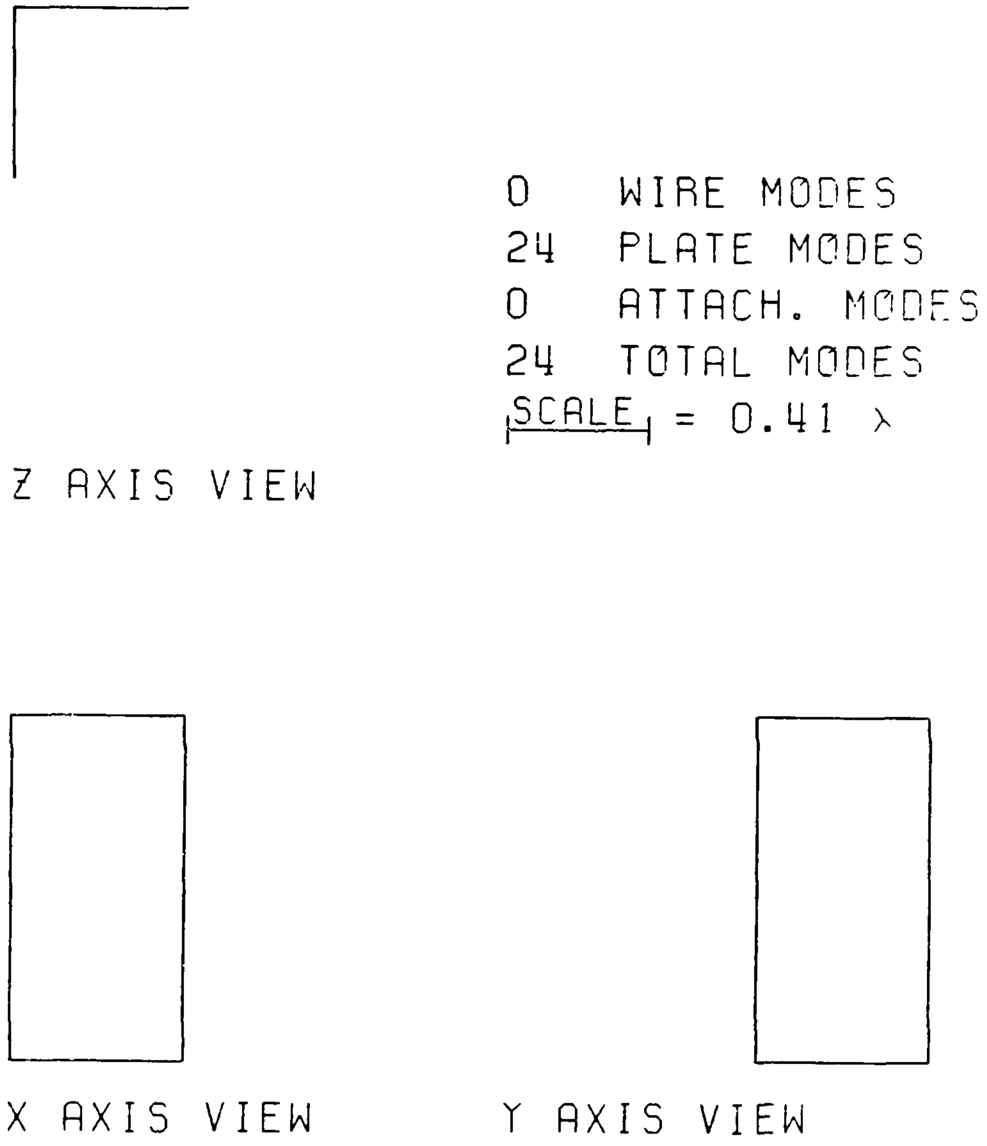


Figure 3-13. A three-view plot of the geometry of Example 2.

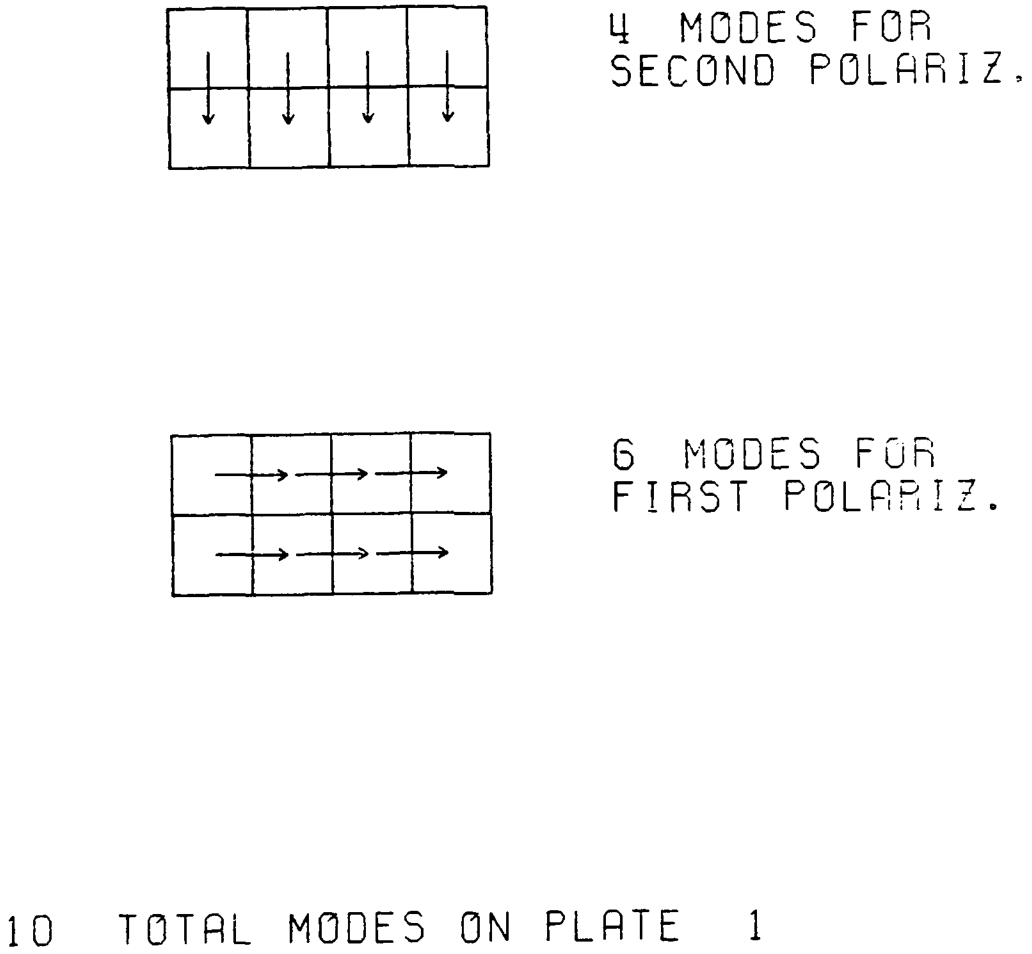


Figure 3-14. (a) Modal outlay on plate 1 of the corner reflector of Example 1,

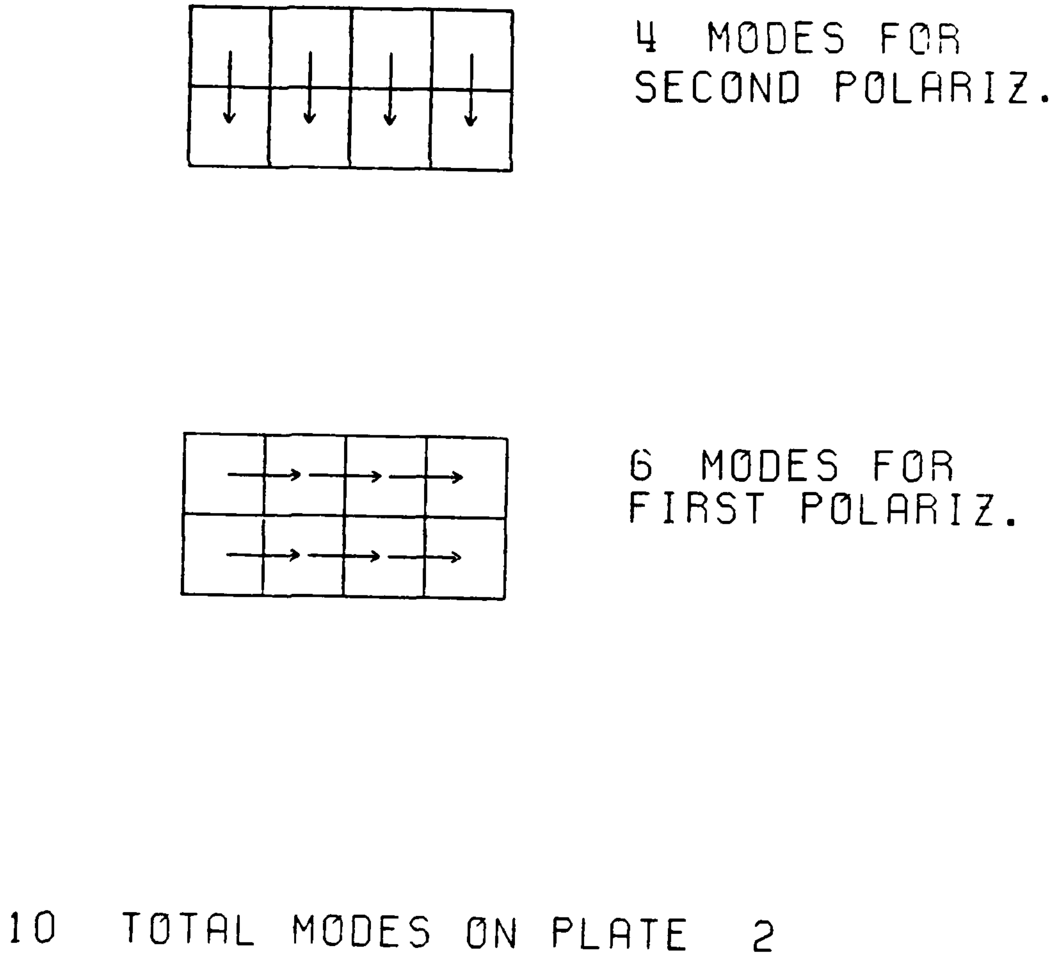


Figure 3-14. (b) Modal outlay on plate 2 of the corner reflector of Example 2,

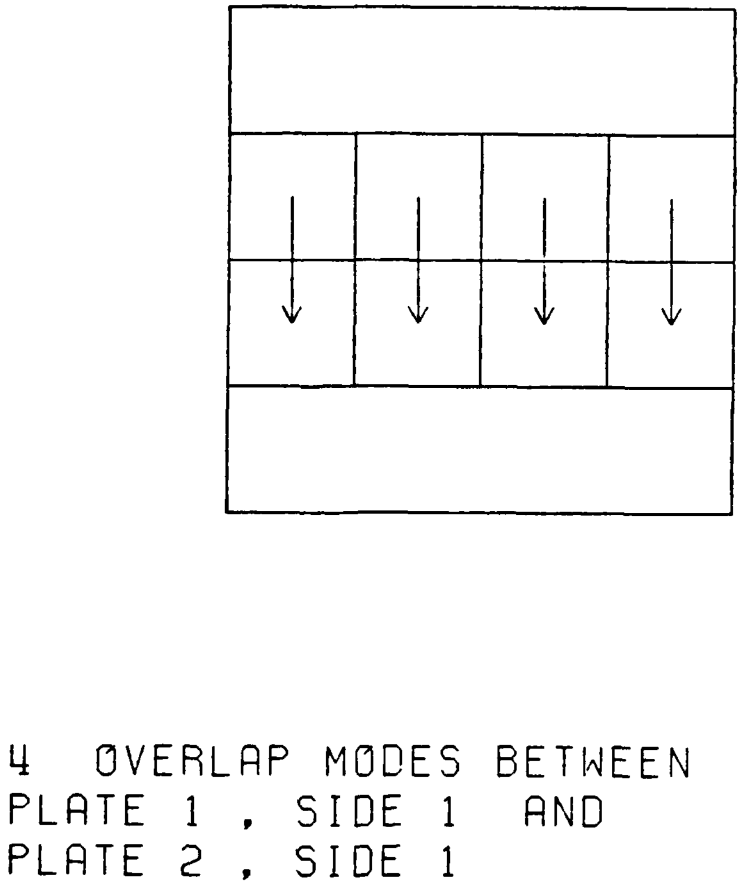


Figure 3-14. (c) modal outlay on the overlap region between plates 1 and 2 of Example 2.

STTM = scattering cross section with incident and scattered fields theta polarized.

SPPM = scattering cross section with incident and scattered fields phi polarized.

STPM = scattering cross section with incident field θ polarized and scattered field ϕ polarized.

SPTM = scattering cross section with incident field ϕ polarized and scattered field θ polarized.

Both the magnitude (in db/λ^2) and the phase (in degrees) are given. Figures 3-15a and b are the polar plots of the magnitudes of STTM and SPPM, respectively.

3. DESIGN EXAMPLE 3

The bistatic scattering pattern in the azimuth plane from the corner reflector of example 2 is examined. This calculation is specified by setting ISA = 2 and THSA = 90.0 in the input file. The incident field is coming from THIN = 90.0 and PHIN = 45.0. The input file for this problem is:

```
READ 1----1 2 1 1 0 0 4 8 18 0 1 1
READ 2----0 1 3 0.0
READ 3----0 1 3 90.0
READ 4----2 1 3 90.0 90.0 45.0
READ 5----0 1 3 90.0
READ 6----300.0 -1.0 0.001
```

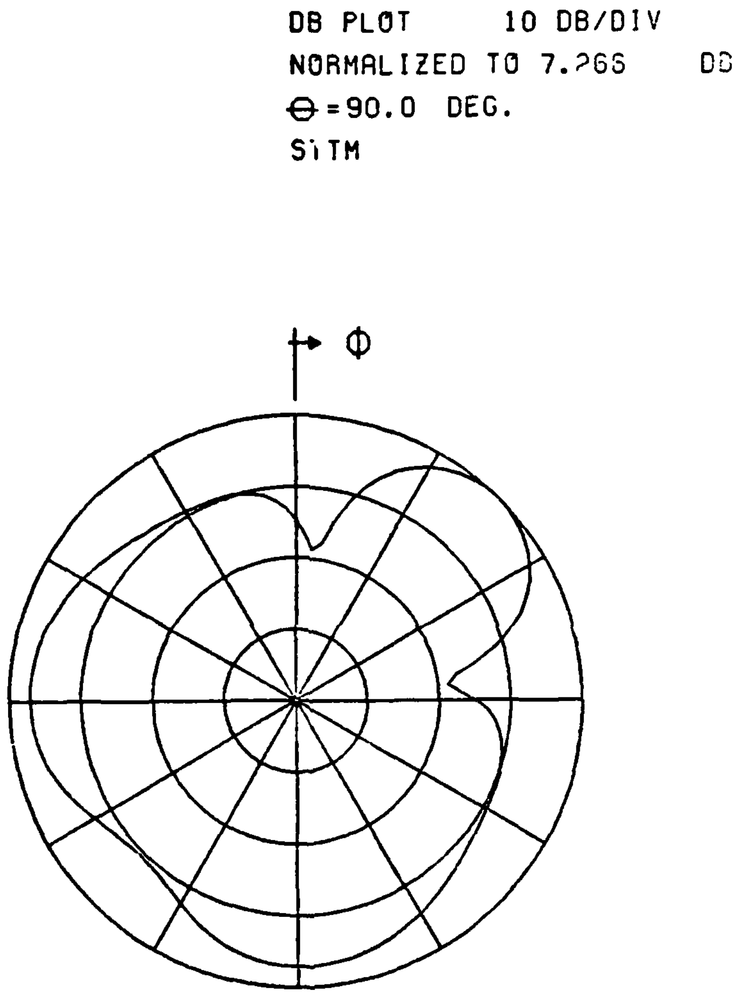


Figure 3-15. (a) θ polarized azimuth backscattering pattern for Example 2,

DB PLOT 10 DB/DIV
NORMALIZED TO 5.791 DB
Θ=90.0 DEG.
SPPM

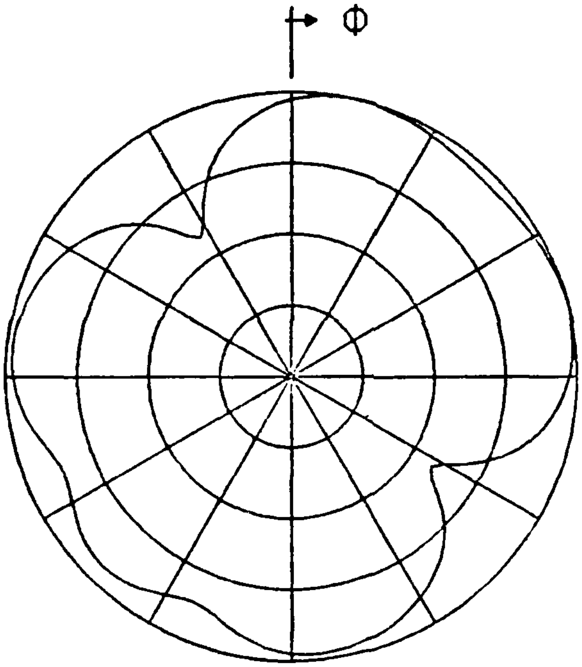


Figure 3-15. (b) ϕ polarized azimuth backscattering pattern for Example 2.

```
READ 7----2
READ 8----4 0.25 1 3 0
READ 9----0.0 0.0 0.0
          0.0 0.0 1.0
          0.5 0.0 1.0
          0.5 0.0 0.0
READ 8----4 0.25 1 3 0
READ 9----0.0 0.0 0.0
          0.0 0.0 1.0
          0.0 0.5 1.0
          0.0 0.5 0.0
READ 10---1 0 .
```

The output of this problem (with IWR = 0 and IWRZT = 0) is shown in Appendix 3. Figures 3-16a and b are the polar plots of the magnitude of STTM and SPPM, respectively.

4. DESIGN EXAMPLE 4

This example will illustrate the use of the READ 10 statement to save CPU time for the impedance matrix calculation. Consider the problem of calculating the input impedance of a quarter-wave monopole at two locations on plate 2 of of the three-plate bend shown in Figure 3-17a. Location 1 is at (x,y,z) = (0.0,0.0,0.0) and location 2 is at (x,y,z) = (0.0,0.3,0.0). The input file for this problem is:

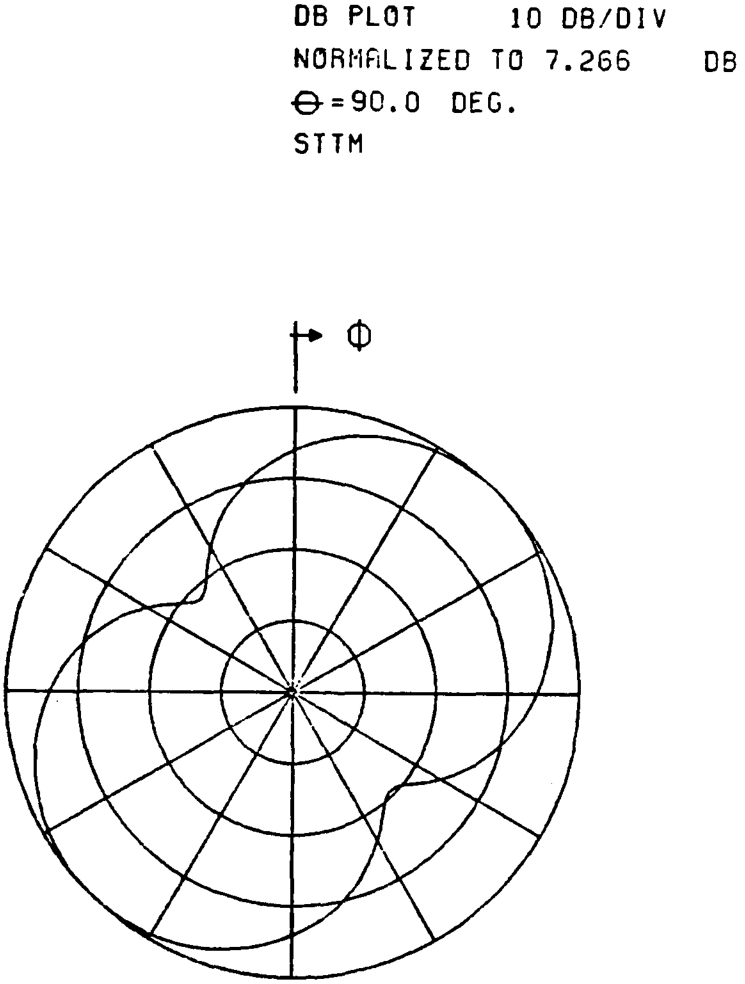
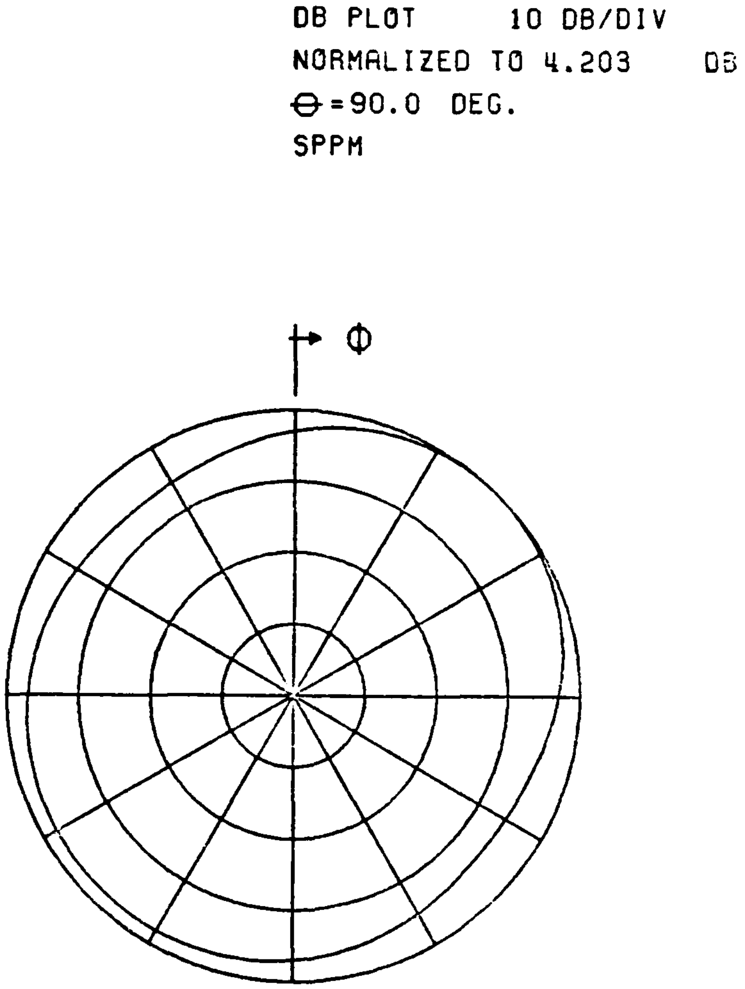


Figure 3-16. (a) θ polarized azimuth bistatic scattering pattern for Example 3,



Fiugre 3-16. (b) ϕ polarized azimuth bistatic scattering pattern for Example 2.

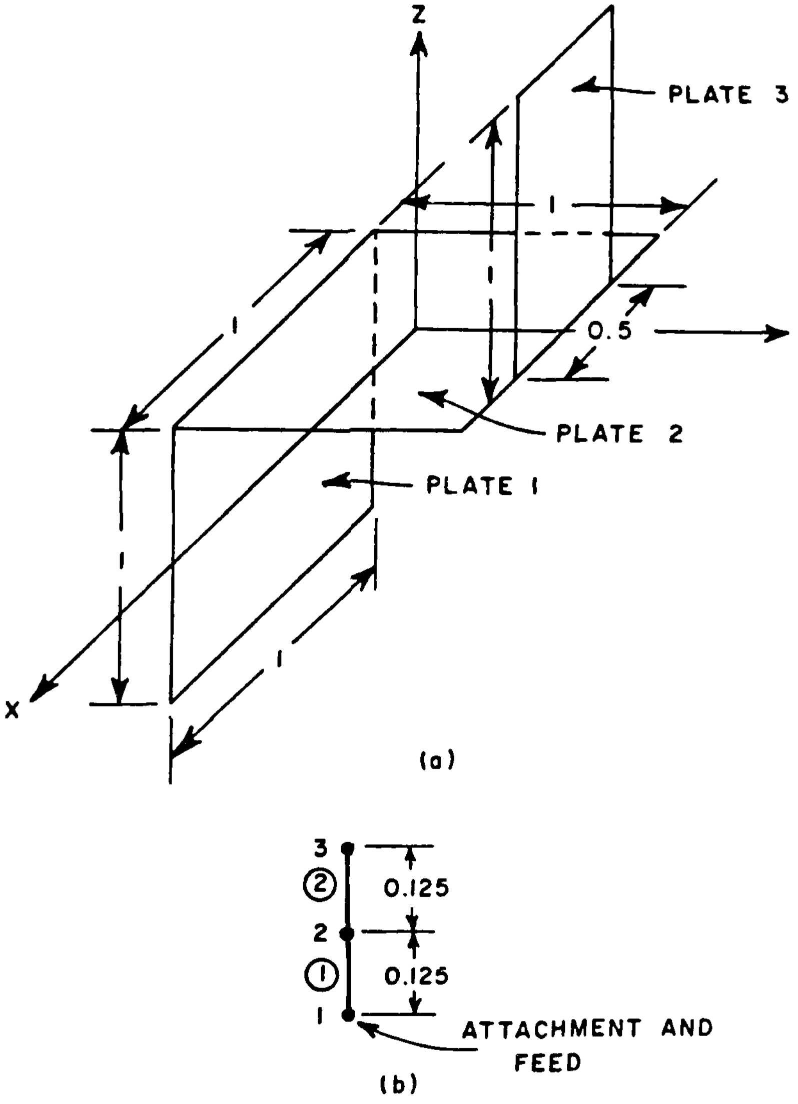


Figure 3-17. (a) Geometry for the three-plate bend of Example 4, (b) Geometry for the wire of Example 4.

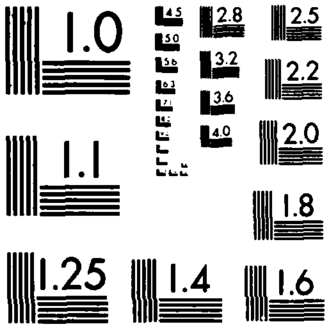
```
READ 1----1 2 1 2 0 0 4 8 18 1 1 0

READ 2----0 1 3 0.0

READ 3----0 1 3 90.0

READ 4----0 1 3 90.0 90.0 45.0

READ 5----0 1 3 90.0

READ 6----300.0 -1.0 0.001

READ 7----3

READ 8----4 0.25 1 3 0

READ 9----0.5 -0.5 -1.0

          0.5 -0.5 0.0

          -0.5 -0.5 1.0

          -0.5 -0.5 0.0

READ 8----4 0.25 1 3 0

READ 9----0.5 -0.5 0.0

          0.5 0.5 0.0

          -0.5 0.5 0.0

          -0.5 -0.5 0.0

READ 8----4 0.25 1 3 0

READ 9----0.25 0.5 0.0

          -0.25 0.5 0.0

          -0.25 0.5 1.0

          0.25 0.5 1.0

READ 10---1 0

READ 11---2 3 1 0 0 0
READ 12---0.0 0.0 0.0
          0.0 0.0 0.125
          0.0 0.0 0.25
READ 13---1 2
          2 3
READ 15---1 0 2 (1.0,0.0) (0.0,0.0) 0.2
READ 10---0 2
READ 11---2 3 1 0
READ 12---0.0 0.3 0.0
          0.0 0.3 0.125
          0.0 0.3 0.25
READ 13---1 2
          2 3
READ 15---1 0 2 (1.0,0.0) (0.0,0.0) 0.2 .
```

Note that there is no READ 14 statement since there are no loads or generators in the wire, except the ones at the attachment point.

To find the input impedance at these two locations one sets NWG = 2 indicating that there are two seperate wire geometries. For the first wire geometry one sets IWRZM = 1 and IRDZM = 0. Thus the entire impedance matrix will be calculated and then written into file ZMAT.DAT on logical unit 1. For the second geometry IRDZM = 2, indicating that the impedance matrix is to be read in from file ZMAT.DAT and that the P/P block of the matrix is not to be recomputed, since the plate geometry has not changed. IWRZM is set to 0 or 1, depending on whether

```
READ 1----1 2 1 2 0 0 4 8 18 1 1 0

READ 2----0 1 3 0.0

READ 3----0 1 3 90.0

READ 4----0 1 3 90.0 90.0 45.0

READ 5----0 1 3 90.0

READ 6----300.0 -1.0 0.001

READ 7----3

READ 8----4 0.25 1 3 0

READ 9----0.5 -0.5 -1.0

          0.5 -0.5 0.0

          -0.5 -0.5 1.0

          -0.5 -0.5 0.0

READ 8----4 0.25 1 3 0

READ 9----0.5 -0.5 0.0

          0.5 0.5 0.0

          -0.5 0.5 0.0

          -0.5 -0.5 0.0

READ 8----4 0.25 1 3 0

READ 9----0.25 0.5 0.0

          -0.25 0.5 0.0

          -0.25 0.5 1.0

          0.25 0.5 1.0

READ 10---1 0

READ 11---2 3 1 0 0 0
```

AD-A135 837   A USER'S MANUAL FOR ELECTROMAGNETIC SURFACE PATCH (ESP)   2/4
CODE VERSION II P..(U) OHIO STATE UNIV COLUMBUS
ELECTROSCIENCE LAB E H NEWMAN ET AL. SEP 83
UNCLASSIFIED ESL-712692-3 N00014-78-C-0049   F/G 20/14   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

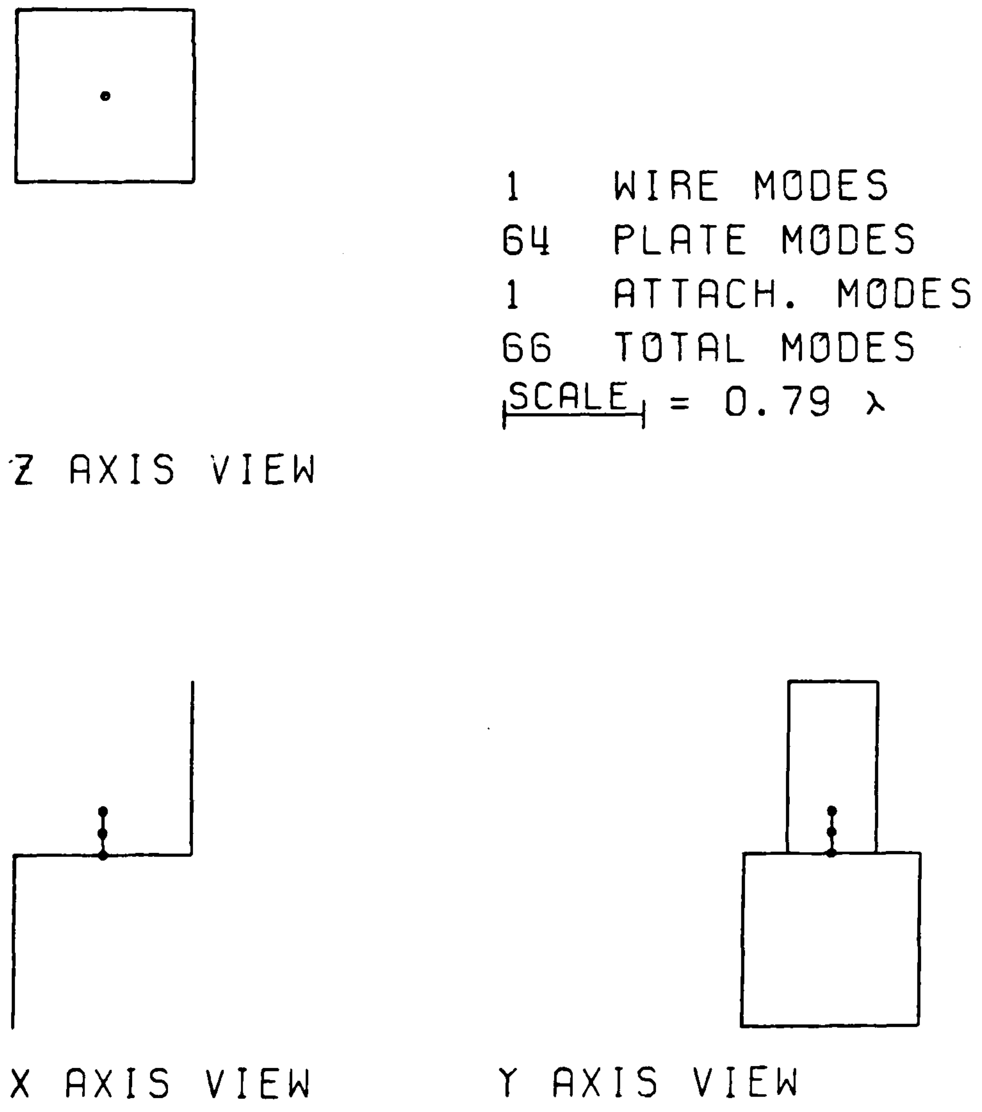


Figure 3-18. A three-view plot of the geometry of Example 4.

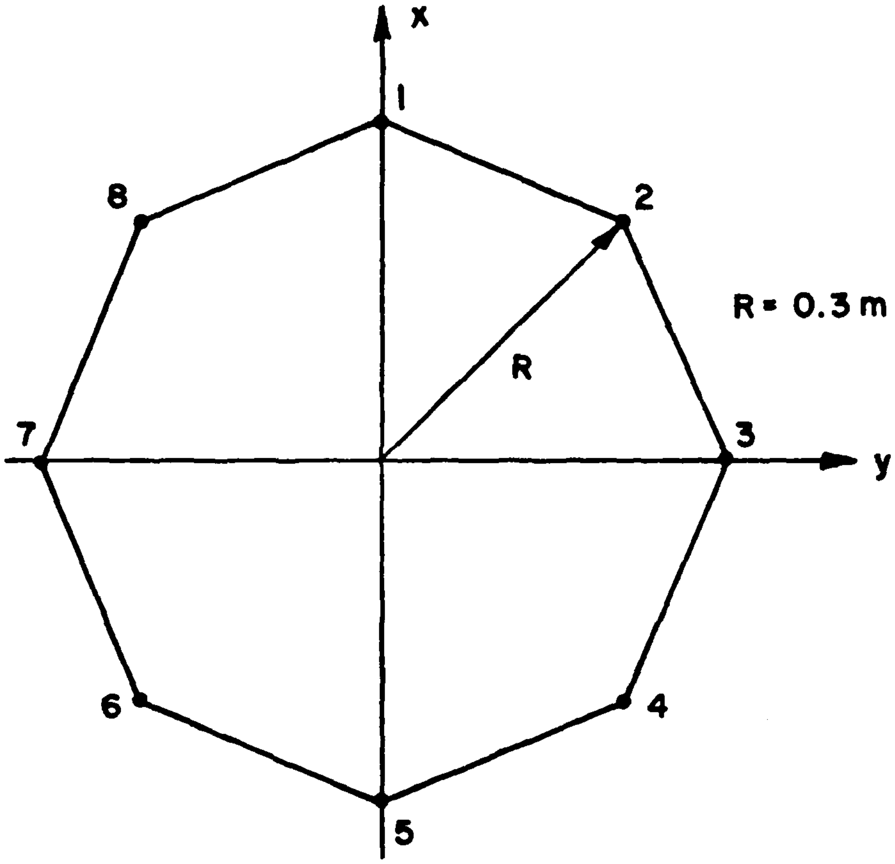


Figure 3-19. Geometry for the polygon plate of Example 5.

```
          0.0 -0.3 0.0
          -0.212 -0.212 0.0
          -0.3 0.0 0.0
          -0.212 0.212 0.0
     READ 10---1 0 .
```

Since IWR = 1, a detailed plot of the modal layout on the plate is obtained (Figure 3-20). Figure 3-21 is the three-view orthographic plot of the regular octagon of Example 5. The output for this problem is shown in Appendix 5.

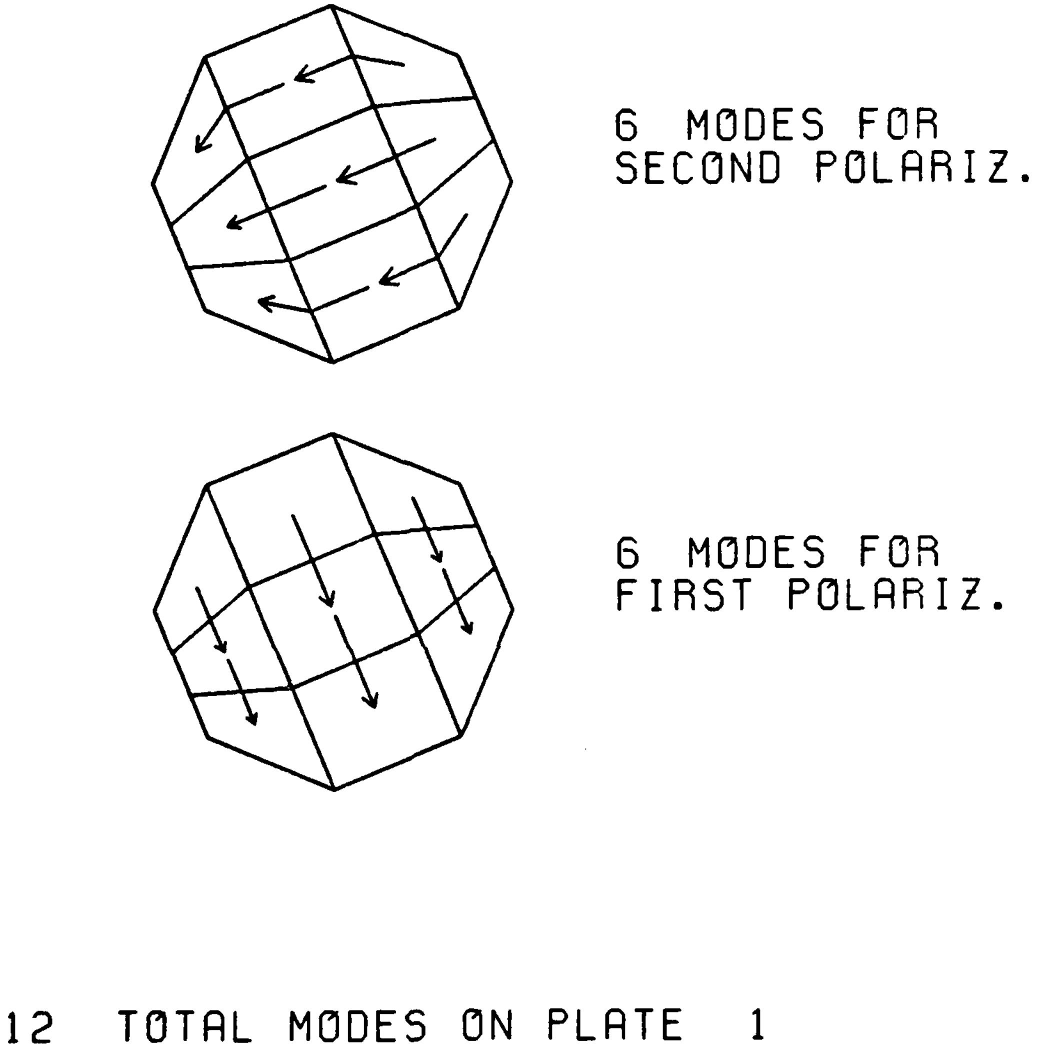


Figure 3-20. Modal layout on the plate of Example 5.

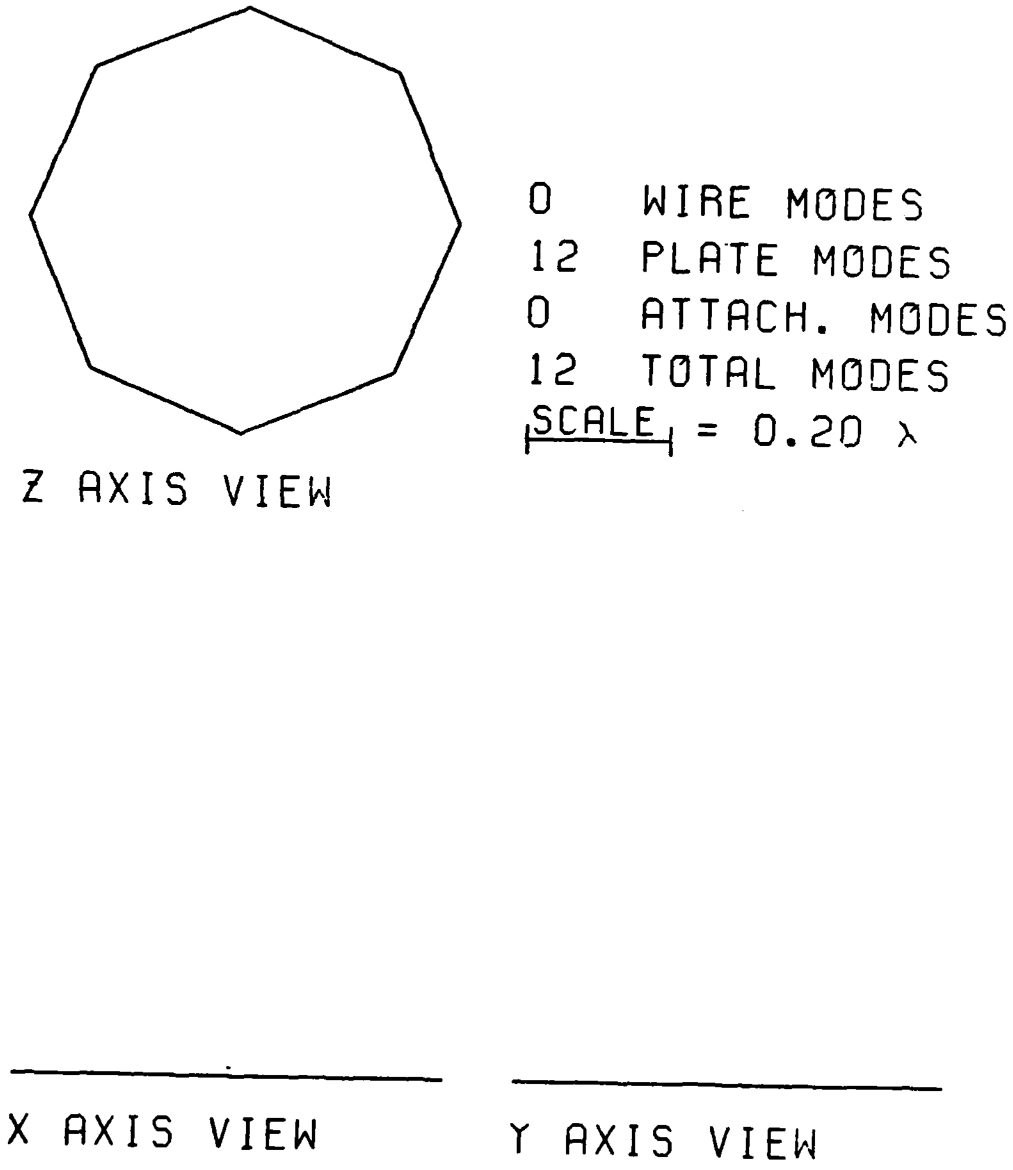


Figure 3-21. Three-view orthographic plot of the geometry of Example 5.

CHAPTER IV

SUBROUTINE DESCRIPTIONS

Several parameters dealing with the geometrical features of the modes are used in more than one subroutine. For reasons of clarity and brevity they are described in detail here, and when they appear later in a subroutine only a brief title description will be given.

A. GENERAL PARAMETERS

Three types of modes are used in this code; wire dipole, surface patch dipole and attachment dipole. Each dipole mode is composed of two monopoles (see Figure 4-1). A wire dipole consists of two wire monopoles, a surface patch dipole consists of two surface patch monopoles and an attachment dipole consists of a wire monopole and a circular disk monopole. The mutual impedance between two dipole modes is the sum of four monopole-to-monopole impedances. In order to evaluate a particular monopole-to-monopole impedance the code needs to know the type of monopoles involved.

IOP = test monopole type indicator.

= 1 implies test monopole is a surface patch.

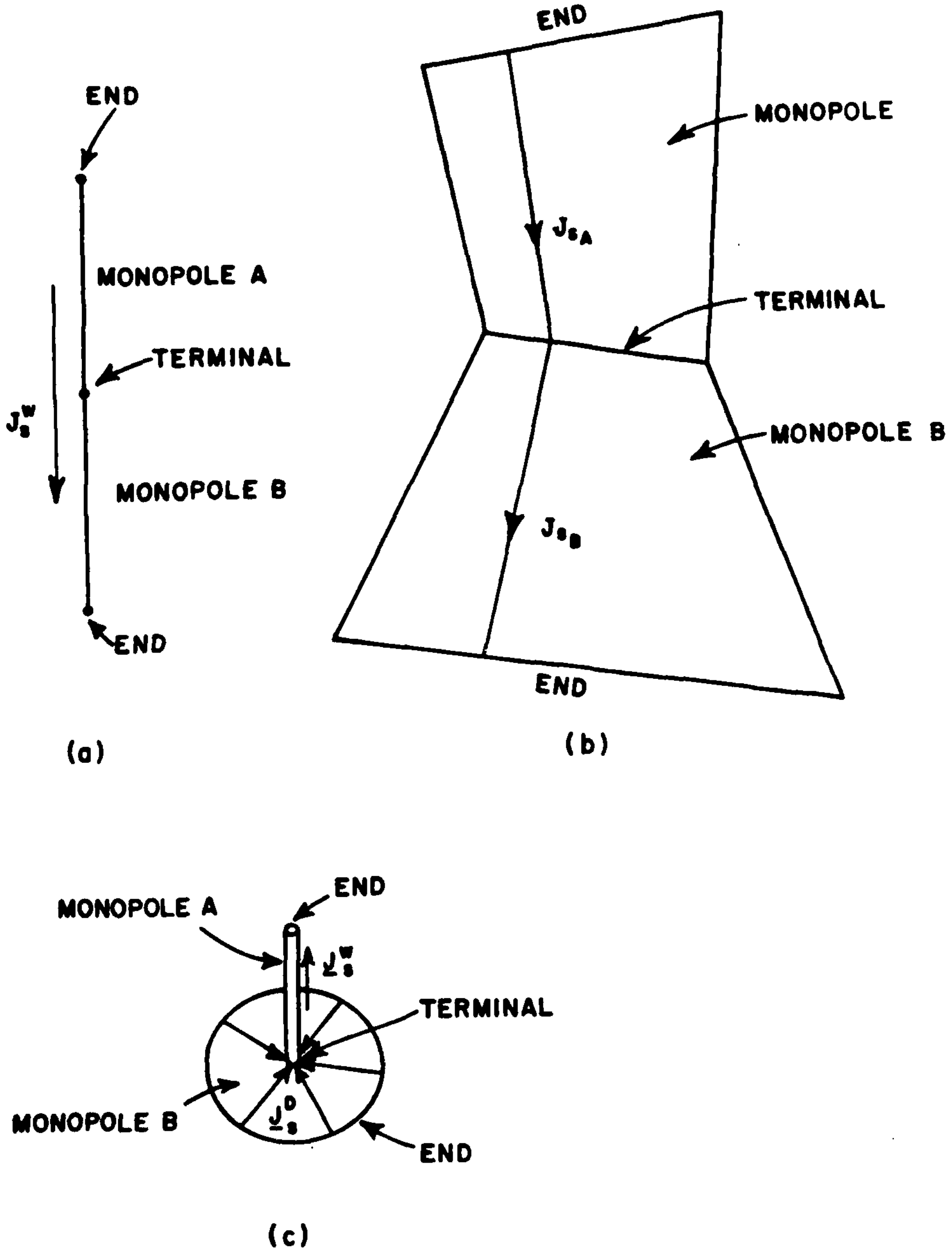


Figure 4-1. (a) Wire dipole mode, (b) arbitrary quadrilateral surface patch dipole mode, and (c) wire attachment dipole mode.

= 2 implies test monopole is a disk.

= 3 implies test monopole is a wire.

= expansion monopole type indicator.

= 1 implies expansion monopole is a surface patch.

= 2 implies expansion monopole is a disk.

= 3 implies expansion monopole is a wire.

IM12 = polarity indicator for the direction of current flow on a test monopole. Consider the three types of modes depicted in Figure 4-1 with their terminals and ends clearly identified. The arrows indicate the direction of positive current flow on the modes.

= 1 implies positive current flows from terminal to end.

= -1 implies positive current flows from end to terminal.

IN12 = polarity indicator for the direction of current flow on an expansion monopole.

= 1 implies positive current flows from terminal to end.

= -1 implies positive current flows from end to terminal.

For monopoles A of the dipoles in Figures 4-1(a), 4-1(b), 4-1(c), IM12 (or IN12) is -1, -1 and +1, respectively. For monopoles B of the same dipoles, IM12 (or IN12) is +1, +1 and -1, respectively.

To accurately represent the current density on a plate we need two orthogonal current polarizations. For a quadrilateral plate it is often sufficient to place modes to cover only one current polarization. For example, consider two overlapping quadrilateral plates such that their touching sides coincide with the overlap segment, i.e., the segment common to both plates. One can place modes on the plates to cover only the polarization parallel to the overlap segment and let the overlap modes cover the other polarization.

IPN(J) = 0 implies place no modes on plate J.

= 1 implies place modes on plate J to cover the first current polarization only.

= 2 implies place modes on plate J to cover the second current polarization only.

= 3 implies place modes on plate J to cover both current polarizations.

For a non rectangular plate IPN(J) = 3.

B. WIRE PARAMETERS

Every wire dipole mode is composed of two wire segments or monopoles. A wire segment is defined by the x,y,z coordinates of its two endpoints as shown Figure 4-2. The first endpoint to be defined is A and the second one is B.

IA(I) = the number of point A of segment I.

IB(I) = the number of point B of segment I.

A wire dipole mode is composed of two wire segments or monopoles (see Figure 4-3).

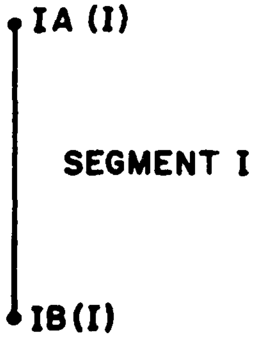


Figure 4-2. Wire segment I.

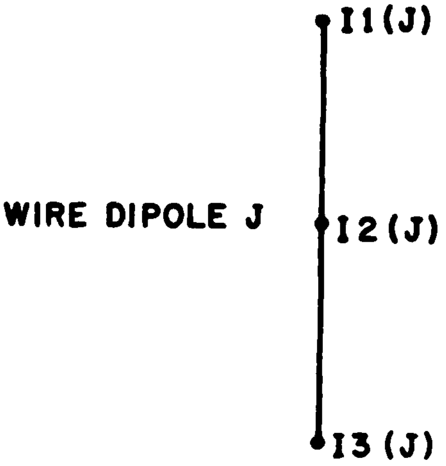


Figure 4-3. Wire dipole mode J.

I1(J) = number of endpoint 1 of wire dipole mode J.

I2(J) = number of terminal point of wire dipole mode J.

I3(J) = number of endpoint 2 of wire dipole mode J.

JA(J) = first segment of wire dipole mode J.

JB(J) = second segment of wire dipole mode J.

MD(J,K) = array containing list of wire dipole modes sharing wire segment J.

All dipoles modes used in this code are sinusoid[illegible]volving the free space propagation constant GAM which in general [illegible]ined as follows:

$$\text{GAM} = \sqrt{(\sigma + j\omega\varepsilon)(j\omega\mu)}$$

but $\sigma = 0$, $\varepsilon = \varepsilon_0$, $\mu = \mu_0$ in this work; so

$$\text{GAM} = -j\omega \sqrt{\mu_0\varepsilon_0} \quad . \tag{4.1}$$

ETA = complex impedance of the medium

$$\text{ETA} = \sqrt{j\omega\mu/\sigma + j\omega\varepsilon}$$

but $\sigma = 0$, $\varepsilon = \varepsilon_0$, $\mu = \mu_0$ in this work; so

$$\text{ETA} = \sqrt{\mu_0/\varepsilon_0} \quad . \tag{4.2}$$

D(I) = length of wire segment I.

SGD(I) = sinh(GAM*D(I)) .

CGD(I) = cosh(GAM*D(I)).

Whenever we include the generators or loads of the wire structure of an antenna it is necessary to know at which wire segment and by what endpoint they are located. If a generator or load is by endpoint A of segment J then its "location" on the wire structure is J. If a generator or load is located by endpoint B of segment J then its "location" on the wire structure is J + NM, where NM is the total number of wire segments.

C. PLATE MODE PARAMETERS

A general plate mode is composed of two arbitrary quadrilateral surface patch monopoles as shown in Figure 4-1. Two types of surface patch monopoles are used in this code; a rectangular surface patch and a quadrilateral (but not rectangular) surface patch (see Figure 4-4).

The following parameters deal with the geometrical features of the surface patch monopole:

IACM = monopole shape indicator for identifying the type of monopoles of a particular test plate dipole mode.

= -3 implies both monopoles of the dipole mode are rectangular surface patches.

= 0 implies either or both monopoles of the dipole mode are quadrilateral surface patches.

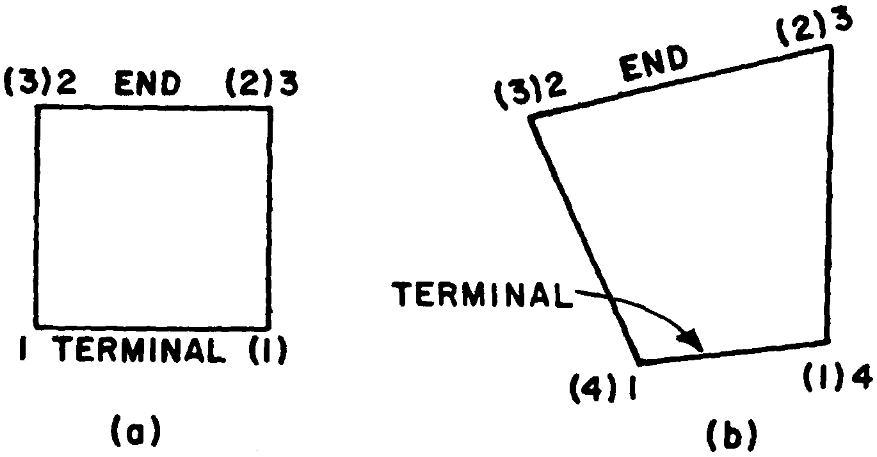


Figure 4-4. (a) Rectangular surface patch monopole, and (b) quadrilateral surface patch monopole.

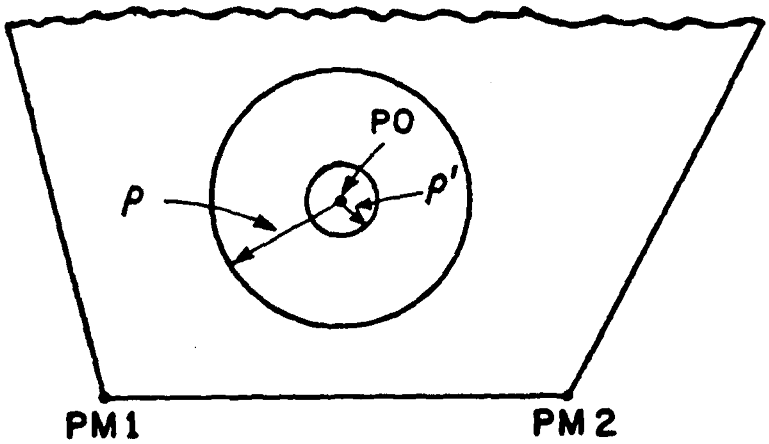


Figure 4-5. Disk monopole on a plate; BDSK=ρ and A=ρ'.

IACN = monopole shape indicator for identifying the type of monopoles of a particular expansion plate dipole mode.

= -3 implies both monopoles of the dipole mode are rectangular surface patches.

= 0 implies either or both monopoles of the dipole mode are quadrilateral surface patches.

A rectangular monopole is defined by three consecutive corners as shown in Figure 4-4(a). The number in parantheses indicates an equivalent way of defining the monopole. In either case side 2-3 is the end side and point 1 is on the terminal side. Positive current flows from the terminal to the end side. A quadrilateral surface patch monopole is defined by the x,y,z coordinates of its four corners as shown in Figure 4-4(b). Note that side 1-2 is the terminal side and side 2-3 is the end side. Again positive current flows from the terminal side to the end side.

D. ATTACHMENT MODE PARAMETERS

An attachment mode is composed of a wire monopole and a disk monopole. The wire monopole is defined by the x,y,z coordinates of its two endpoints as described for a wire segment above.

The disk monopole is defined (see Figure 4-5) by the x,y,z coordinates of its center and two points on its plane. Normally those two points are the first two corners of the plate the disk lies on.

The following parameters deal with the geometry of the disk monopole:

BDSK(I) = outer radius of the disk monopole I. From now on the term disk monopole I will imply the disk monopole of attachment dipole mode I.

A = inner radius of the disk monopole. Normally A is the wire monopole radius.

The following parameters deal with the electric field of a disk monopole parallel to a surface patch monopole:

ERVSR(K,JJ) = array containing values of the radial component of the electric field of disk monopole K versus the radial distance ρ.

RMIN(K) = the minimum distance between the center of disk monopole K and any point on the surface patch monopole.

RMAX = the maximum distance from the disk monopole center to any point on the surface patch monopole.

DR(K) = the increment in the value of ρ, i.e., ρ = RMIN(K) + DR(K)*JJ where JJ is the JJ-th point the electric field is evaluated.

DIST = the distance between the planes of the disk and surface patch monopoles.

The term attachment or wire attachment point implies the point where a wire segment attaches to a plate. The number of attachment points is the same as the number of attachment dipole modes.

E. NOTES ON THE IMPEDANCE MATRIX

The general expression for an impedance matrix element is given by Equation (2.9). Since every test and expansion mode is made up of two monopoles the mutual impedance between two dipole modes is the sum of four monopole-to-monopole impedances. In particular

$$Zmn = Z^{t_1e_1} + Z^{t_1e_2} + Z^{t_2e_1} + Z^{t_2e_2}$$

where t_1 and e_1 refer to monopole 1 of the test and expansion dipole modes, respectively. Similarly, t_2 and e_2 refer to monopole 2 of the test and expansion modes, respectively.

When full surface patch test dipole modes are used (IFIL = 0), the impedance matrix is symmetric because the test and expansion modes are identical (Galerkin's method). Because of the symmetry only the lower triangular part of the matrix is calculated. An impedance matrix element Zmn is stored in the linear array ZT(K) at location K = (n-1)(NT) -(n -n)/2 + m. NT is the total number of dipole modes in the problem.

If the surface patch test modes are represented as single filaments (IFIL = 1), the impedance matrix is no longer symmetric and the whole matrix has to be calculated. An impedance matrix element Zmn is stored in the two dimensional array ZTF(M,N).

F. MISCELLANEOUS NOTES:

A common block defining several parameters used by most of the subroutines in the code is defined in the main program. It has the following form:

```
COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
```

where

WV = wavelength in meters.

PI = 3.141592

A = wire radius in meters.

Q = 0.001*WV.

GAM = complex free space propagation constant (0.0 - j2*PI/WV).

ETA = complex free space impedance (376.7 + j0.0).

XK = free space wavenumber (2*PI/WV).

The following are general comments about the program:

1. All values of lengths and distances are in meters (m) unless otherwise noted.
2. All values of angles are in degrees unless otherwise noted.
3. Many numerical integrations are done using Simpson's rule integration. The number of Simpson's rule integration intervals , specified with such parameters as INTP, NPT, INTD and NINT, should always be an even number.

4. All the subroutines explained in this chapter are listed in the Appendix section in the order that they appear here. The subroutine listings are Appendices 6 through 40.

5. The far zone electric field of a mode is a function of the spherical coordinates r, θ and ϕ. In particular, $E(r,\theta,\phi)$ can be written as

$$E(r,\theta,\phi) = \frac{e^{-jkr}}{r} E_f(\theta,\phi) \quad .$$

6. Whenever the far field of a monopole is mentioned in this code it is assumed that we mean $E_f(\theta,\phi)$ and that the exp(-jkr)/r dependence is supressed.

7. The expressions for the near zone fields of PWS monopole do not include the contributions from the point or line charges at the endpoints of the monopole, since these charges disappear when two monopoles are connected to form a dipole.

The following subroutines dealing with wire monopoles are included with permission of Professor J. H. Richmond:

SORT,SGANT,CBES,DSHELL,GGS,GGMM,EXPJ,GANT1,SQROT,GFF .

They are documented in

Richmond, J. H., "Computer Program for Thin-Wire Structures in a Homogeneous Conducting Medium", Report 2902-12, August 1973, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering, Columbus, Ohio.

1. SUBROUTINE PLPLCK

PURPOSE:

Subroutine PLPLCK checks every plate to ensure that all of its corners lie on the same plane.

GENERAL FORM:

PLPLCK(PCN,ICN,IPL,NC,TOUCH,NP,IOK) .

THE FOLLOWING ARE INPUTS:

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of plate K.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

NC = number of plate corners.

TOUCH = touch indicator. If the distance of a plate corner from the plane of the plate (defined below in the NOTES section) is greater than TOUCH/2 then the geometry of the plate is being defined incorrectly in the INPUT file.

NC = plate number.

THE FOLLOWING ARE OUTPUTS:

IOK = status indicator. If IOK = 0 the run is aborted and PLPLCK returns to the main program.

NOTES:

Subroutine PLPLCK defines the x-y plane by corners 1,2 and NC of the plate. Then for every corner IC 1,2,NC its distance ZP from the x-y plane is evaluateo. If ZP is larger than TOUCH/2, the run is aborted and an error message is printed. If ZP is less than TOUCH/2, corner IC is redefined as the projection of the old IC on the x-y plane.

2. SUBROUTINE PLATE3

PURPOSE:

Subroutine PLATE3 generates the modal layout on a polygonal plate. The only restriction is that the plate does not have more than one interior angle greater than 180 degrees.

GENERAL FORM:

```
PLATE3(PC,NC,ICN,NP,NDNPLT,PA,PB,IPLM,SEGM,IQUAD,WV,IRE,IP,MPL1,
MPL2,IOK,NM12,NM23,IGS) .
```

THE FOLLOWING ARE INPUTS:

PC(I,J) = x,y,z coordinates (I = 1,2,3) of the J-th plate corner.

NC = number of plate corners.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

NP = plate number.

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

SEGM = the maximum permissible segment size for a surface patch monopole side. Normally SEGM = 0.25*WV for accurate calculation of the impedance matrix elements.

WV = wavelength.

IRE = rectangular/polygonal plate indicator.

= 0 implies the plate is polygonal.

= 1 implies the plate is rectangular.

IP = polarization indicator, same as IPN(NP).

IGS = generating plate side indicator. If IGS is an integer greater than 0 but less than NC, it specifies the number of the plate side to be used as the generating side. If IGS = 0, then PLATE3 chooses the largest plate side as the generating side.

THE FOLLOWING ARE OUTPUTS:

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the J-th corner of monopole A of the I-th plate mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the J-th corner of monopole B of the I-th plate mode.

IQUAD(I) = monopole shape indicator for identifying the type of monopoles of plate mode I.

MPL1 = the total number of modes covering the first polarization.

MPL2 = the total number of modes on the plate.

IOK = status indicator. If IOK = 0, the run is aborted and an error message is printed.

NM12 = the number of segments on the 1-2 side of a rectangular plate (if IRE = 1).

NM23 = the number of segments on the 2-3 side of a rectangular plate (if IRE = 1).

NOTES:

Consider the arbitrary polygon plate shown in Figure 4-6. First, PLATE3 checks the number of interior angles larger than 180 degrees and also finds the length DMX of the longest side ISMX with endpoints IC and IC2. If there is more than one interior angle greater than 180 degrees, PLATE3 returns to the main program.

Side ISMX is the generating side. PLATE3 moves from IC until it finds the next corner different from IC2 and repeats the same procedure from corner IC2. The resulting quadrilateral is defined by corners IC,IC2,B,A. Let NAS and NBS be the required number of segments along DA and DB, respectively. If NAS -NBS is larger than MDM, a constant specified at the beginning of PLATE3, then either side DA or DB is made shorter until NAS -NBS is less than MDM. The purpose of this test is to minimize the number of modes on the plate. This procedure is repeated until the polygon plate is broken into quadrilaterals. Triangles are

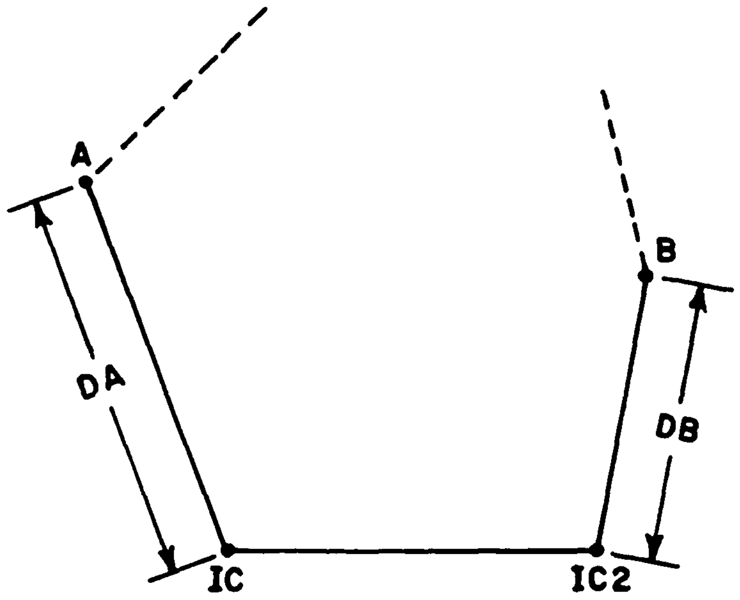


Figure 4-6. A three-side section of an arbitrary polygonal plate.

treated as quadrilaterals by defining a fourth corner at the midpoint of their longest side.

Finally, the modes along the direction of ISMX are defined followed by the orthogonal set of modes.

3. SUBROUTINE POPLOV

PURPOSE:

Whenever two or more plates are touching, overlap modes are needed to allow for a continuous current at the plate-to-plate junctions. Subroutine POPLOV sets up those overlap modes and for a multiplate junction finds the minimum linearly independent set of overlap modes.

GENERAL FORM:

```
POPLOV(NPLTS,PCN,NCNRS,TOUCH,SEGM,PA,PB,NOVT,NPLTM,IPL,IPLM,ICN,
IOVT, DOVL, ITK, NOPL, IQUAD, WV,NDNPLT,OVEP) .
```

THE FOLLOWING ARE INPUTS:

NPLTS = the total number of plates.

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.

NCNRS(I) = the number of corners on plate I.

TOUCH = touch parameter. If the separation between any two sides of two plates is less than TOUCH, the the plates are considered to be overlapping. Normally TOUCH = 0.001*WV.

SEGM = the maximum permissible segment size for a surface patch monopole side. Normally SEGM = 0.25*WV for accurate calculation of the impedance matrix elements.

NPLTM = total number of plate modes.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

NDNPLT(I) = total number of plate modes through plate I.

WV = wavelength.

THE FOLLOWING ARE OUTPUTS:

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith surface patch mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith surface patch mode.

NOVT = total number of overlap modes.

IOVT(I,J) = indicator for identifying the plates or common sides of the overlap plate pair I.

IOVT(I,1) = plate A of pair I.

IOVT(I,2) = side A of pair I.

IOVT(I,3) = plate B of pair I.

IOVT(I,4) = side B of pair I.

DOVL(I) = length in meters of the segment common to both plates of the overlap pair I.

ITK(I) = number of overlap modes in overlap pair I.

NOPL = total number of overlap plate pairs.

IQUAD(J) = monopole shape indicator for identifying the type of surface patch monopoles of overlap plate mode J.

For more explanations refer to Figure 4-7.

NOTES:

Subroutine POPLOV is divided into three sections. Section 1 (lines 18 through 107) determines the existence of all overlap plate pairs. An overlap pair is defined by two touching plates as if those were the only overlapping plates. Section 1 calculates and stores the parameters NOPL, IOVT, DOVL, OVEP and ITK.

Section 2 (lines 110 through 319) eliminates unnecessary overlap modes by checking for linear dependencies. All plate overlap pairs along the same overlap line are compared against its others three at a time. Overlap pair I is defined in DO 170 loop, overlap pair J is defined in DO 175 loop and overlap pair K is defined in DO loop 180.

The first criterion for a linear dependency is that plate A, side A of pair I be the same as plate A, side A of pair J; plate B, side B of pair I be the same as plate A, side A of pair K; and plate B, side B of pair J be the same as plate B, side B of pair K.

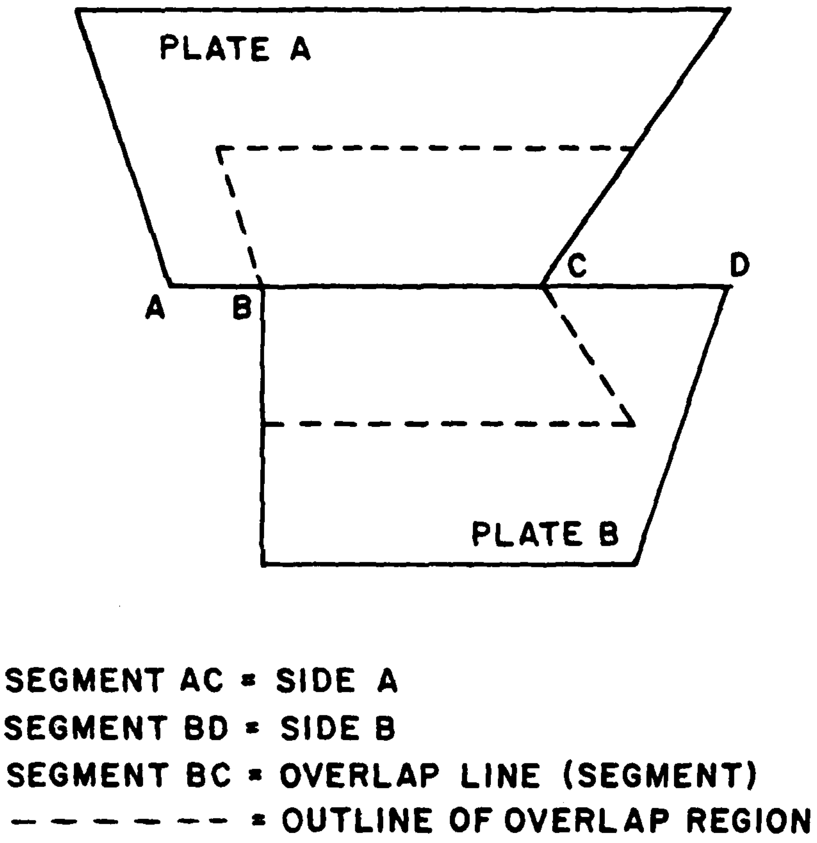


PLATE A
C
D
A
B
PLATE B
SEGMENT AC = SIDE A
SEGMENT BD = SIDE B
SEGMENT BC = OVERLAP LINE (SEGMENT)
- - - - - - - = OUTLINE OF OVERLAP REGION

Figure 4-7. The overlap region between touching polygonal plates.

The second criterion for linear dependency is that the overlap modes of at least two of the three overlap pairs must mesh together. If the overlap segment of all three pairs is the same and at least two of the three overlap pairs contain the same number of modes, a linear dependency exists and one overlap pair can be eliminated. If that test fails, then in order to have a linear dependency at least two of the three overlap mode lengths must be the same and the endpoints of each overlap segment must coincide with an endpoint of an overlap mode in the other two overlap pairs. An overlap mode length is defined as the overlap segment length divided by the number of overlap modes. If it is determined that one of the pairs can be eliminated, say I, then ITK(I) = 0.

Section 3 (lines 320 through 377) constructs the overlap modes. The coordinates of the modes are stored in arrays PA and PB and the monopole shape indicators for every mode are stored in array IQUAD.

4. SUBROUTINE FGPOV

PURPOSE:

Subroutine FGPOV finds the four points that define the overlap regions on both plates (See Figure 4-7) of an overlap plate pair. All overlap modes lie within those regions. FGPOV also finds NOV, the minimum number of overlap modes needed to cover that region.

GENERAL FORM:

```
FGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCNRS,PCN,ICN,IPL,SEGM,NDNPLT,PA,PB,
IPLM,WV,TOUCH,TPSI,SPSI, OMSP,NOV) .
```

THE FOLLOWING ARE INPUTS:

NPA = plate A of the overlap plate pair.

ISA = side A of the overlap plate pair.

NPB = plate B of the overlap plate pair.

ISB = side B of the overlap plate pair.

OE(I,J) = x,y,z coordinates (I = 1,2,3) of the two endpoints (J = 1,2) of the common overlap segment.

DOV = length of the common overlap segment.

NCNRS(I) = number of corners on plate I.

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

SEGM = the maximum permissible segment size for a surface patch monopole side. Normally SEGM = 0.25*WV for accurate calculation of the impedance matrix elements.

NDNPLT(I) = total number of plate modes through plate I.

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith surface patch monopole.

PB(I,J,K) = x,y,z coordinates (K =1,2,3) of the Jth corner of monopole B of the Ith surface patch monopole.

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

WV = wavelength.

TOUCH = touch parameter. If the separation between any two sides of two plates is less than TOUCH, the plates are considered to be overlapping. Normally TOUCH = 0.001*WV.

TPSI = cos(ψ)

SPSI = sin(ψ) (Refer to Figure 4-8).

THE FOLLOWING ARE OUTPUTS:

OMSP(I,J,K) = x,y,z coordinates (I = 1,2,3) of the four corners (J = 1,2,3,4) defining the overlap region on plate NPA(K = 1) or NPB(K = 2).

NOV = number of overlap modes between plate NPA and NPB.

NOTES:

First, preliminary values of OMSP are found from the plate corners and calls to subroutine FMDC. In most cases the final value for two of the OMSP points on a plate are given by the overlap segment endpoints. For the remaining points consider the case in Figure 4-8 where OMSP(I,4,1) is determined. The preliminary value of OMSP(I,4,1) is given by point 1. If OE(I,1) is also a corner of plate A or if the

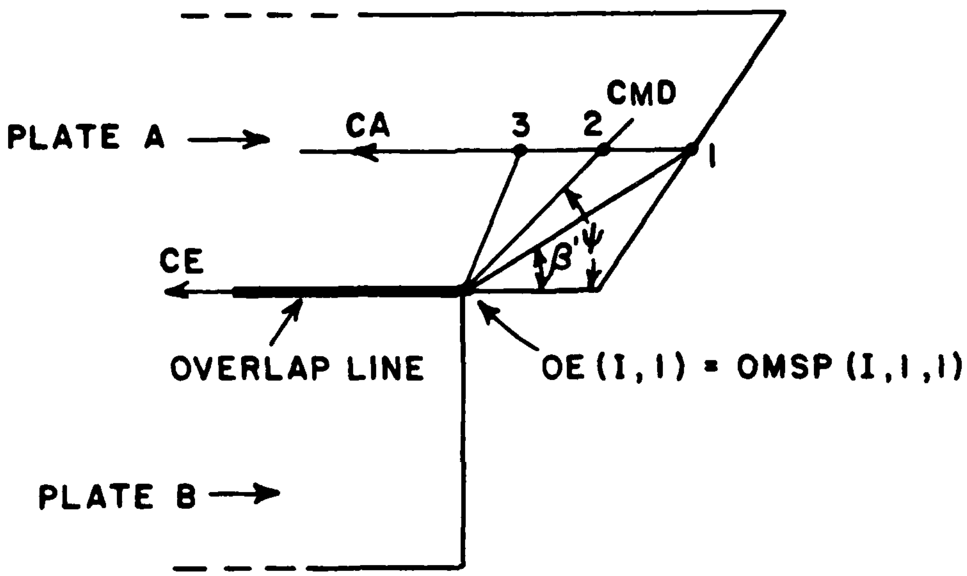


Figure 4-8. Definition of the overlap region corners.

angle β' is greater than ψ and the distance between point 1 and OE(I,1) is less than SEGM, then OMSP(I,4,1) is point 1.

Otherwise OMSP(I,4,1) is moved along the vector CA until it reaches point 2 at the line CMD, which is at an angle ψ with the overlap line. If the distance between point 2 and OE(I,1) is larger than SEGM, OMSP(I,4,1) continues moving along vector CA until the distance between OMSP(I,4,1) and OE(I,1) is equal to SEGM (see point 3). All remaining OMSP points are found using the same procedure.

5. SUBROUTINE FMDC

PURPOSE:

Subroutine FMDC finds the preliminary value of the overlap region corner OMSP(I,MC,IAB) which is point 1 of Figure 4-8.

GENERAL FORM:

FMDC(NDNPLT,PCN,ICN,IPL,PA,PB,IPLM,NC,NAC,MC,NP,IAB,CE,TOUCH,OMSP,
WV,NPO) .

THE FOLLOWING ARE INPUTS:

NDNPLT(I) = Total number of plate modes through plate I.

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith surface patch monopole.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith surface patch monopole.

IPLM = dimension indicator for the maximum number of plate mode used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

NC = plate corner that lies on the overlap line.

NAC = plate corner adjacent to NC but not on the overlap line.

MC = overlap region corner to be defined.

NP = plate number.

IAB = plate A (IAB = 1) or plate B (IAB = 2).

CE = directional cosine of the overlap line.

TOUCH = touch parameter. If the separation between any two sides of two plates is less than TOUCH, the plates are considered to be overlapping. Normally TOUCH = 0.001*WV.

NPO = number of the other plate in the overlap pair.

For further explanation see Figure 4-9.

THE FOLLOWING ARE OUTPUTS:

OMSP(I,J,K) = x,y,z coordinates (I = 1,2,3) of the four corners (J = 1,2,3,4) of the overlap region on plate NPA (K = 1) or NPB (K = 2).

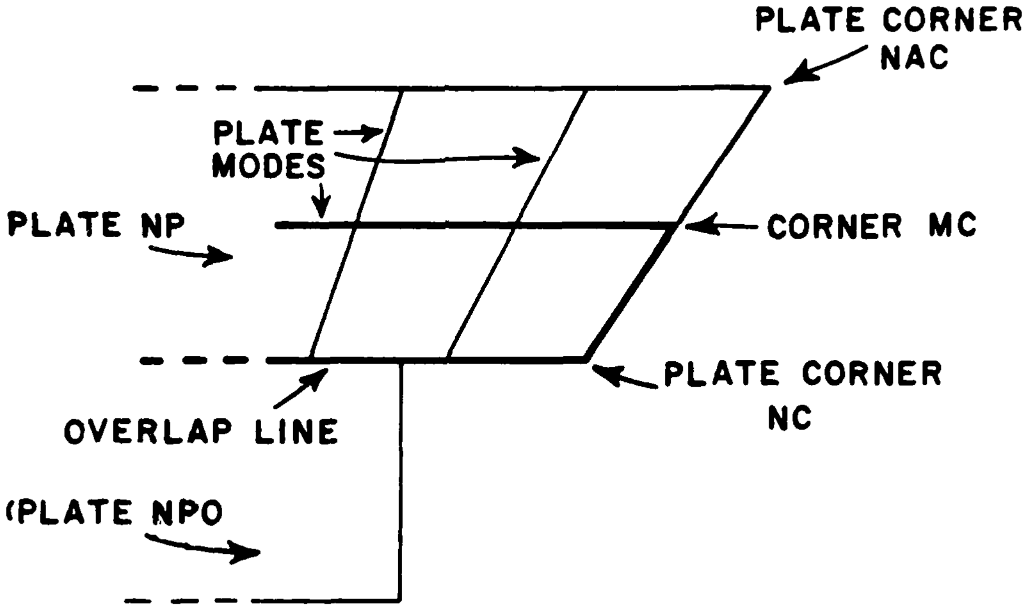


Figure 4-9. Definition of the preliminary overlap region corners.

NOTES:

Subroutine FMDC searches through all of the modes on plate NP and determines the mode which minimizes the distance between corner 2 or 3 of the monopole A or B and the plate corner PCN(I,NC,NP). OMSP(I,MC,IAB) is then given by corner 1 or 4 of the same monopole.

6. SUBROUTINE MOPLOT

PURPOSE:

Subroutine MOPLOT gives an orthographic plot of two touching plates and of the overlap modes existing between the two plates. The plot indicates the plate numbers and the plate side numbers of the sides along the overlap line. The orthographic plot is what one would see if he unfolded the two plates so that they lie on the same plane.

GENERAL FORM:

MOPLOT(PCN,NCNRS,IPL,ICN,PA,PB,IPLM,IOVT,ITK,NOPL,NPLTM, NOVT)

THE FOLLOWING ARE INPUTS:

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.

NCNRS(I) = number of corners on plate I.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith mode.

IPLM = dimension indicator for maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

IOVT(I,J) = indicator for identifying the plates or common sides of the overlap plate pair I.

IOVT(I,1) = plate A of pair I.

IOVT(I,2) = side A of pair I.

IOVT(I,3) = plate B of pair I.

IOVT(I,4) = side B of pair I.

ITK(I) = number of overlap modes in group I.

NOPL = total number of plate overlap plate pairs.

NPLTM = total number of plate modes.

NOVT = total number of overlap plate modes.

See Figure 3-14(c) for an example.

7. SUBROUTINE MPLOT

PURPOSE:

Subroutine MPLOT plots the modal layout of a particular plate as it is defined by subroutine PLATE3.

GENERAL FORM:

MPLOT(NCNRS,PCN,NPL,ICN,IPL,IPLM,NPL11,NPL22,NDNPLT, PA,PB,IPN)

THE FOLLOWING ARE INPUTS:

NCNRS(I) = the number of corners of plate I.

PCN(K,I,J) = x,y,z coordinates (K=1,2,3) of the Ith corner of the Jth plate.

NPL = the total number of plates.

ICN = dimension indicator for maximum number of plate corners used in array PCN(3,ICN,IPL).

IPL = dimension indicator for maximum number of plates used in array PCN(3,ICN,IPL).

IPLM = dimension indicator for maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

NPL11(I) = the total number of modes covering the first polarization on plate I.

NPL22(I) = the total number of modes on plate I.

NDNPLT(I) = the total number of modes through plate I.

PA(I,J,K) = x,y,z coordinates (K=1,2,3) of Jth corner of monopole A of the Ith plate dipole mode.

PB(I,J,K) = x,y,z coordinates (K=1,2,3) of Jth corner of monopole B of the Ith plate dipole mode.

IPN(I) = polarization indicator for plate I.

NOTES:

Subroutine MPLOT draws two outlines of the plate in consideration on the same page. Then using arrays PA and PB it draws the modal grid on both outlines. The bottom grid represents the modes covering the other polarization. Each mode is identified by drawing an arrow from monopole A to monopole B of the mode.

Also on the same page MPLOT outputs the total number of modes for every polarization and the total number of modes on the whole plate. See Figure 3-14(a) and (b)for an example.

8. SUBROUTINE GPLOT2

PURPOSE:

Subroutine GPLOT2 gives an othographic plot of the antenna or scatterer geometry. In particular, it gives a projected view of the geometry as seen along the x, y and z axis. Plate sides are shown in solid lines and wire segments are shown as solid lines with small circles at the endpoints. GPLOT2 also gives a summary of the wire, plate and attachment modes of the geometry as well as a scale indicating what one inch is in wavelengths.

GENERAL FORM:

```
GPLOT2(NM,NP,X,Y,Z,IA,IB,NPLTS,PCN,IPL,NWR,NPLTM,NAT, WV,ICN,NCNRS)
```

THE FOLLOWING ARE INPUTS:

NM = the total number of wire segments.

NP = the total number of wire points.

X(I),Y(I),Z(I) = x,y,z coordinates of the Ith wire point.

IA(J) = endpoint A of wire segment J.

IB(J) = endpoint B of wire segment J.

NPLTS = the total number of plates.

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the J-th corner of plate K.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

NWR = total number of wire modes.

NPLTM = total number of plate modes.

NAT = total number of wire attachments.

WV = wavelength.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

NCNRS(I) = number of corners of plate I.

See Figure 3-13 for an example.

9. SUBROUTINE ZTOT

PURPOSE:

Subroutine ZTOT evaluates the Moment Method impedance matrix elements Zmn, described by Equation (2.9). Every Zmn is the sum of four monopole-to-monopole impedances which are calculated in ZTOT by calling the appropriate monopole-to-monopole impedance subroutines.

If full test surface patch monopoles are used (IFIL = 0), then the impedance matrix can be visualized as shown below

$$\left[\begin{array}{c|c|c} W/W & W/P & W/A \\ P/W & P/P & P/A \\ A/W & A/P & A/A \end{array}\right]$$

W = Wire

P = Plate

A = Attachment .

Subroutine ZTOT evaluates only the lower triangular part of the matrix and stores the entries in the linear array ZT(MN) where MN = (N-1)*NTOT -(N*N-N)/2.0 + M.

If filamentary test surface patch monopoles are used (IFIL = 1), then the impedance matrix reduces to

$$\left[\begin{array}{c|c} W/W & W/A \\ A/W & A/A \end{array}\right] .$$

In this case the whole matrix is evaluated and its entries are stored in the two dimensional array ZTF(M,N).

GENERAL FORM:

ZTOT(IA,IB,INM,I1,I2,I3,JA,JB,,MD,NWR,ND,NM,NP,CGD,SGD,D,X,Y,Z ZLD,
NPL,NAT,ZS,IRDZM,ZLDA,PA,PB,NSA,NPLA,PCN,IPL,IPLM,BDSK,ZT,ZTF,NM12N,
NM23N,ICN,NDNPLT,NOVT,INT,INTP,INTD,CMM,ERVSR,RMIN,DR,IAT,IPN,
IQUAD,NCNRS,IFIL,IREC, ICC) .

THE FOLLOWING ARE INPUTS:

IA(I) = endpoint A of wire segment I.

IB(I) = endpoint B of wire segment I.

INM = dimension indicator for array MD(INM,4).

I1(J) = endpoint 1 of wire dipole J.

I2(J) = terminal point of wire dipole J.

I3(J) = endpoint 2 of wire dipole J.

JA(J) = segment A of wire dipole mode J.

JB(J) = segment B of wire dipole mode J.

MD(J,L) = list of wire dipoles modes sharing wire segment J.

NWR = total number of wire modes.

ND(J) = total number of wire dipoles modes sharing wire segment J.

NM = total number of wire segments.

NP = total number of wire points.

D(J) = length of wire segment J.

CGD(J) = cosh(GAM*D(J)).

SGD(J) = sinh(GAM*D(J)).

X(I),Y(I),Z(I) = x,y,z coordinates of wire point I.

ZLD(II) = complex impedance load at wire "location" II.

NPL = total number of plates.

NAT = total number of wire attachments points.

ZS = complex wire surface impedance.

IRDZM = read indicator.

IRDZM = 0 implies do not read in existing impedance matrix and calculate the whole new impedance matrix.

IRDZM = 1 implies read in the matrix and calculate the whole new impedance matrix except the W/W block.

IRDZM = 2 implies read in the matrix and calculate the whole new impedance matrix except the P/P block.

IRDZM = 3 implies use existing impedance matrix.

ZLDA(K) = complex impedance load at attachment K.

PA(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole A of the Ith surface patch mode.

PB(I,J,K) = x,y,z coordinates (K = 1,2,3) of the Jth corner of monopole B of the Ith surface patch mode.

NSA(K) = the wire segment "location" of wire attachment K.

NPLA(K) = number of plate where wire attachment K is located.

PCN(I,J,K) = x,y,z coordinates (I = 1,2,3) of the Jth corner of the Kth plate.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(IPLM,4,3) and PB(IPLM,4,3).

BDSK(K) = outer radius of disk monopole of Kth attachment mode.

NM12N(J) = the total number of modes in the 1-2 direction on rectangular plate J.

NM23N(J) = the total number of modes in the 2-3 direction on rectangular plate J.

ICN = dimension indicator for the maximum numbuer of plate corners used in array PCN(3,ICN,IPL).

NDNPLT(J) = total number of plate modes through plate J.

NOVT = total number of overlap modes.

INT = number of Simpson's rule intevals used in the filament to filament integrations.

INTP = number of Simpson's rule intervals used in integrating over the surface patch expansion monopoles.

INTD = number of Simpson's rule integration intervals used in integrating over the disk expansion monopoles.

CMM = wire conductivity in megamhoms/meter. If CMM = -1.0, this implies perfect conductivity.

ERVSR(K,JJ)= array containing the values of the ρ component of the electric field of disk monopole K versus the radial distance ρ.

RMIN(K) = the minimum value of ρ corresponding to ERVSR(K,1).

DR(K) = the increment in the value of ρ, i.e., ρ = RMIN(K) + DR(K)*JJ.

IAT = dimension indicator for the maximum number of attachments used in array ERVSR(IAT,400).

IPN(NPL) = polarization indicator for plate NPL.

IQUAD(J) = monopole shape indicator for identifying the type of surface patch monopoles composing plate mode J.

NCNRS(NPL) = total number of corners on plate NPL.

IFIL = indicator for choosing either full surface patch test monopoles (IFIL = 0) or filamentary surface patch test monopoles (IFIL = 1).

IREC(I) = rectangular/polygonal plate indicator.

= 0 implies plate I is polygonal (not rectangular).

= 1 implies plate I is rectangular.

ICC = Dimension indicator for array ZTF.

THE FOLLOWING ARE OUTPUTS:

ZT(MN) = complex impedance linear array used when IFIL = 0.

ZTF(M,N) = complex impedance matrix used when IFIL = 1.

NOTES:

Because of the importance of subroutine ZTOT it is included in Appendix (14) with extensive comments.

10. SUBROUTINE TOPO

PURPOSE:

The impedance matrix for a single rectangular plate has a great deal of Toeplitz properties. Subroutine TOPO identifies impedance elements which are equal (within a +, -sign) by virtue of the Toeplitz properties.

GENERAL FORM:

TOPO(NM12,NM23,K,L,MT,NT,SGN) .

THE FOLLOWING ARE INPUTS:

NM12 = the number of segments in the 1-2 direction.

NM23 = the number of segments in the 2-3 direction.

K = local row number of desired impedance element, i.e., as if the first mode of the plate was mode 1.

L = local column number of the desired impedance element.

THE FOLLOWING PARAMETRES ARE OUTPUTS:

MT,NT = local row and column number corresponding to K and L;i.e. entry Z(K,L) = Z(MT,NT)*SGN .

SGN = sign factor (+1.0 or -1.0).

11. SUBROUTINE PLTST2

PURPOSE:

Subroutine PLTST2 calculates the mutual impedance between test monopole M and expansion monopole N. M is always a quadrilateral surface patch monopole and N is either a quadrilateral surface patch or a wire monopole. The current distributions on a surface-patch and a wire monopole are given by Equations (2.13) and (2.11), respectively.

GENERAL FORM:

```
PLTST2(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,XM4,YM4,ZM4,IM12,JOP,
XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,YN4,ZN4,IN12,NPT,NINT,
IACM,IACN,ZMN) .
```

THE FOLLOWING PARAMETERS ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

PM1=(XM1,YM1,ZM1), PM2=(XM2,YM2,ZM2), PM3=(XM3,YM3,ZM3), PM4 =(XM4,YM4,ZM4) = x,y,z coordinates of the four corners of the surface patch test monopole M.

IM12 = polarity indicator for the direction of current flow on the test monopole.

JOP = expansion monopole type indicator.

PN1=(XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3), PN4=(XN4,YN4,ZN4) = x,y,z coordinates of four points of the expansion monopole. If the expansion monopole is a surface patch PN1,PN2,PN3,PN4 are its four corners. If the expansion monopole is a wire PN1,PN2 are its endpoints and PN3,PN4 are not used.

IN12 = same as IM12 but for expansion monopole.

NPT = the number of Simpson's rule integration intervals used in integrating over the test monopole. This numerical integration is implemented in the DO 10 loop.

NINT = the number of Simpson's rule integration intervals used in integrating over the surface patch expansion monopole (if JOP = 1). This integration takes place in subroutine ZWTPE2.

IACM = monopole shape indicator for identifying the type of surface patch monopoles of the particular test plate mode.

IACN = same as IACM but for expansion mode.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between monopole M and monopole N.

NOTES:

Expansion monopole is a surface patch (JOP = 1):

If NPT = 1, both the expansion and test monopoles are represented by one filament. Their mutual impedance is evaluated by a single call to ZWTPE2 and subsequent averaging of this filament to filament impedance, as described by Equations (21) and (22), gives ZMN.

If NPT is larger than 1 the test monopole is represented by NPT + 1 filaments. The mutual impedance of each filament to the expansion monopole is evaluated by ZWTPE2 and all those partial impedances are summed using a Simpson's rule weighting.

Expansion monopole is a wire (JOP = 3):

NPT is always larger than one. The test monopole is represented by NPT + 1 filaments and the mutual impedance between every filament and the wire monopole is evaluated by ZWTWE. The final monopole-to-monopole impedance is the Spline rule summation of the partial filament-to-wire monopole impedances.

12. SUBROUTINE ZWTPE2

PURPOSE:

Subroutine ZWTPE2 calculates the mutual impedance between a wire test monopole M and a quadrilateral surface patch monopole N. The currents on the wire and the polygonal surface patch monopole are given by Equations (2.11) and (2.13), respectively.

GENERAL FORM:

```
ZWTPE2(XM1,YM1,ZM1,XM2,YM2,ZM2,DM,IM12,XN1,YN1,ZN1,XN2,YN2,
ZN2,XN3,YN3,ZN3,XN4,YN4,ZN4,IN12,NPLS,IACN,ZMN,KINT)
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK

PM1=(XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2) = x,y,z coordinates of the two wire monopole endpoints.

DM = length of test wire monopole.

IM12 = polarity indicator for the direction of current flow on the test wire monopole.

PN1=(XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3), PN4=(XN4,YN4,ZN4) = x,y,z coordinates of the four surface patch monopole corners.

IN12 = same as IM12 but for expansion monopole.

NPLS = number of Spline rule integration intervals used in the integration over the plate expansion monopole.

IACN = monopole shape indicator for identifying the type of surface patch monopoles of the particular expansion plate mode.

KINT = indicator for setting the INT integration parameter used for the filament-to-filament impedance calculations in subroutine GGS1. If KINT = 0, then ZWTPE2 assigns the value of INT. If KINT = 1, then INT = 0.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between monopole M and monopole N.

NOTES:

If NPLS = 1, ZWTPE2 evaluates ZMN by calling GGS1 once and modifying the intermediate result using Equations (2.21) and (2.22).

If NPLS is larger than 1 ZWTPE2, represents the expansion monopole N by NPLS filaments and evaluates the mutual impedance between every filament and the wire test monopole using GGS1. Then it sums all those partial impedances by a Spline rule integration weighting to obtain ZMN. A substantial part of the subroutine is spent in deciding the value of INT. If INT = 0, the impedance calculations in GGS1 are done in closed form; If INT = 2 or 4, the calculations are done using 2 or 4 interval Simpson's rule integration, respectively.

13. SUBROUTINE PLTTST

PURPOSE:

Subroutine PLTTST calculates the mutual impedance between test monopole M and expansion monopole N. M is always a rectangular surface patch and N can be a wire, rectangular surface patch or disk monopole with current distributions given by Equations (2.11), (2.12) and (2.16), respectively.

GENERAL FORM:

```
PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,JOP,XN1,YN1,ZN1,
XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,NINT,BN, ZMN)
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK

PM1= (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2),PM3=(XM3,YM3,ZM3) = x,y,z coordinates of three consecutive corners of the rectangular surface patch test monopole.

IM12 = polarity indicator for the direction of current flow on the test monopole.

JOP = expansion monopole type indicator.

PN1= (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3) = x,y,z coordinates of three points on the expansion monopole. If the expansion monopole is a surface patch, PN1,PN2,PN3 are three consecutive corners of the patch. If it is a wire, PN1,PN2 are its endpoints and PN3 is not used. If it is a disk, PN1 is the center of the disk, PN2 and PN3 are two points on the plane of the disk and PN4 is not defined.

IN12 = same as IM12 but for expansion monopole.

INTP = the number of Simpson's rule integration intervals used in integrating over the test surface patch monopole. The integration is implemented in the DO 10 loop.

NINT = the number of Simpson's rule integration intervals used in integrating over the expansion surface patch (JOP = 1) or expansion disk (JOP = 2) monopole. The integration is implemented in subroutines ZWTPE (JOP = 1) or ZWTDE(JOP = 2).

BN = outer radius of expansion disk monopole (if JOP = 2).

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between monopole M and monopole N.

NOTES:

If the expansion monopole is a surface patch, PLTTST checks for parallel current density vectors or parallel vectors transverse to the current density vectors (see Chapter II, section D.2). If either is true, then ZMN is calculated by calling subroutine PPLTS. If neither is true, the test surface patch monopole is represented by INTP+1 filaments and the mutual impedance between every filament and the expansion monopole is evaluated by calling subroutine ZWTPE. The partial filament-to-monopole impedances are summed using a Simpson's rule weighting to give ZMN.

If the expansion monopole is not a plate (JOP = 2 or JOP = 3), then again the test monopole is represented by INTP+1 filaments and the mutual impedance between every filament and the expansion monopole is evaluated by calling the appropriate subroutine; i.e., ZWTDE if JOP = 2 or ZWTWE if JOP = 3. All those partial filament-to-monopole impedances are summed via a Simpson's rule weighting to give ZMN.

14. SUBROUTINE PPLTS

PURPOSE:

Subroutine PPLTS calculates the mutual impedance between two rectangular surface patch monopoles that are parallel in the sense described in Figure 4-10.

GENERAL FORM:

```
PPLTS(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,XN1,YN1,ZN1,XN2,YN2,
ZN2,XN3,YN3,ZN3,IN12,NPT,ZMN) .
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2),PM3=(XM3,YM3,ZM3) = x,y,z coordinates of three consecutive corners of the surface patch test monopole.

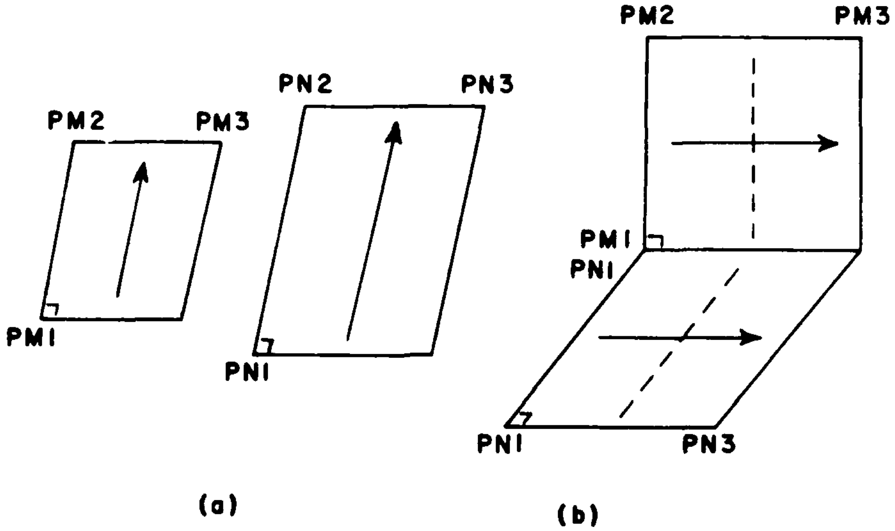


Figure 4-10. (a) Current direction vectors are parallel.
(b) Vectors transverse to the current direction vectors are parallel.

IM12 = polarity indicator for the direction of current flow on the test monopole.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3) = x,y,z coordinates of three consecutive corners of the surface patch expansion monopole.

IN12 = same as IM12 but for the expansion monopole.

NPT = number of Simpson's rule integration intervals used in integrating over the test and expansion monopoles.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the two rectangular surface patch monopoles.

NOTES:

Subroutine PPLTS finds DMIN = the minimum separation and DMAX = the maximum separation between the two surface patch monopoles. DM12 = the length of side PM1-PM2 of the test monopole and DN12 = the length of side PN1-PN2 of the expansion monopole. Then it sets up a wire monopole with length DM12 parallel to a wire monopole with length DN12 and lets their separation D vary between DMIN and DMAX. It calculates the corresponding mutual impedances and stores them in array ZVSD. If a log singularity arises because the separation is too small, it is removed analytically.

Subsequently the two surface patches are represented by filaments and array ZVSD is used to evaluate the filament-to-filament impedances by linear extrapolation. ZMN is the Simpson's rule weighted sum of all the filament-to-filament impedances.

15. SUBROUTINE ZWTPE

PURPOSE:

Subroutine ZWTPE calculates the mutual impedance between a wire test monopole and a rectangular surface patch monopole.

GENERAL FORM:

```
ZWTPE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,
ZN3,IN12,INTP,ZMN) .
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2) = x,y,z coordinates of the test wire monopole endpoints.

IM12 = polarity indicator for the direction of current flow on the test monopole.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3) = x,y,z coordinates of three consecutive corners of the expansion surface patch monopole.

IN12 = same as IM12 but for the expansion monopole.

INTP = the number of Simpson's rule integration intervals used in integrating over the expansion surface patch monopole. The integration is implemented in the DO 10 loop.

THE FOLLOWING ARE OUTPUTS:

ZMN = The mutual impedance between the wire and surface patch monopole.

NOTES:

The surface patch expansion monopole is represented by INTP+1 filaments. The mutual impedance between each filament and the wire test monopole is evaluated by ZGSMM and all those partial impedances are summed with a Simpson's rule weighting to give ZMN.

The following parameters are defined in ZWTPE to be used by subroutine ZGSMM. Refer to Figure 4-11.

DM = the length of test wire monopole.

GAM = the complex free space propagation constant defined by Equation (4.1).

CGDM = cosh(GAM*DM)

SGDM = sinh(GAM*DM)

DN = the length of side 1-2 of the expansion surface patch monopole.

SGDN = sinh(GAM*DN)

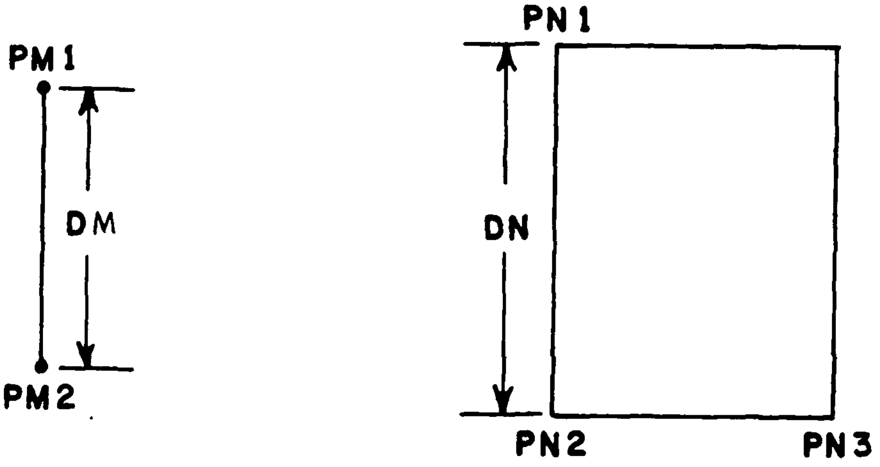


Figure 4-11. Lengths DM and DN as defined in subroutine ZWTPE.

16. SUBROUTINE ZWTDE

PURPOSE:

Subroutine ZWTDE calculates the mutual impedance between a wire test monopole and a disk expansion monopole.

GENERAL FORM:

```
ZWTDE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,XNO,YNO,ZNO,XN1,YN1,ZN1,XN2,YN2,ZN2,
INTD,B,ZMN) .
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2) = x,y,z coordinates of the two test wire monopole endpoints.

IM12 = polarity indicator for the direction of current flow on the test monopole.

PO = (XNO,YNO,ZNO) = x,y,z coordinates of disk monopole center PO.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2) = x,y,z coordinates of two points on the plane of the disk monopole.

INTD = number of Simpson's rule integration intervals used in integrating over the disk expansion monopole. This integration is implemented in the DO 10 loop.

B = outer radius of disk monopole.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the wire test monopole M and the disk expansion monopole N.

NOTES:

The disk monopole is represented by INTD radial filaments. The mutual impedance between every filament and the wire monopole is evaluated by ZGSMM and all those partial mutual impedances are summed by a Simpson's rule weighting to give ZMN.

The following parameters are defined in ZWTDE and used by subroutine ZGSMM. Refer to Figure 4-12.

GAM = complex free space propagation constant defined by Equation (4.1).

DM = the length of the test wire monopole.

DN = the difference between the disk outer and inner radius.

SGDM = sinh(GAM*DM) .

CGDM = cosh(GAM*DM) .

SGDN = sinh(GAM*DN) .

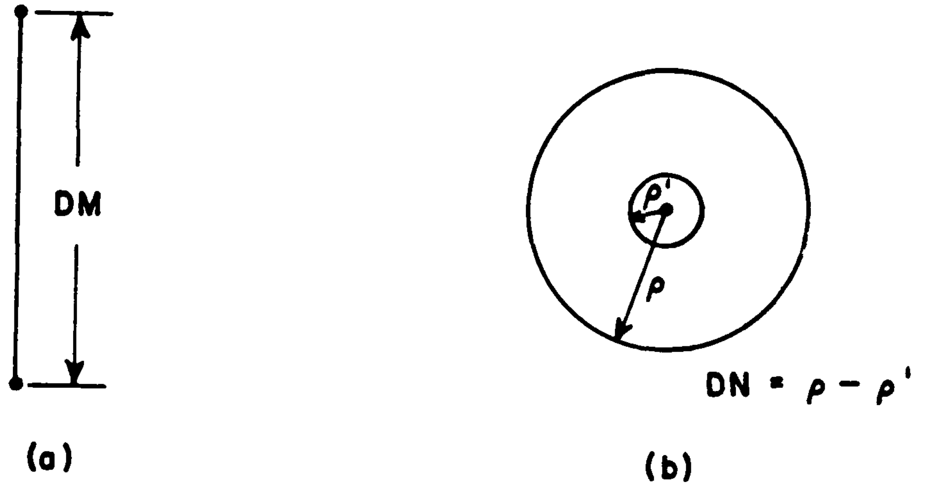


Figure 4-12. Lengths DM and DN as defined in Subroutine ZWTDE.

17. SUBROUTINE ZWTWE

PURPOSE:

Subroutine ZWTWE calculates the mutual impedance between a wire test monopole M and a wire expansion monopole N. First it sets up certain geometric parameters dealing with both monopoles. Then it calls subroutine ZGSMM to evaluate the mutual impedance between the two wire monopoles.

GENERAL FORM:

```
ZWTWE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,XN1,YN1,ZN1,XN2,YN2,ZN2,IN12,ZMN,IWW)
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK

PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2) = x,y,z coordinates of the test wire monopole endpoints.

IM12 = polarity indicator for the direction of current flow on the test monopole.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2) = x,y,z coordinates of the expansion wire monopole endpoints.

IN12 = same as IM12 but for the expansion monopole.

IWW = indicator for choosing the minimum distance between two filaments which share one or more points. IWW = 1 implies the minimum separation is A, the wire radius. IWW = 0 implies the minimum separation is Q, normally chosen as 0.001*WV.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the wire test monopole M and the wire expansion monopole N.

NOTES:

The following parameters defined in ZWTWE are used by ZGSMM which ZWTWE calls.

GAM = complex free space propagation constant defined by Equation (4.1).

DM = the length of the test wire monopole.

DN = the length of the expansion wire monopole.

SGDM = sinh(GAM*DM) .

CGDM = cosh(GAM*DM) .

SGDN = sinh(GAM*DN) .

18. SUBROUTINE ZGSMM

PURPOSE:

Subroutine ZGSMM calculates the mutual impedance between two PWS filaments.

GENERAL FORM:

```
ZGSMM( XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,A,D1,CGD1,SGD1,D2,SGD2,Z12)
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

PA = (XA,YA,ZA),PB=(XB,YB,ZB) = x,y,z coordinates of the test PWS monopole endpoints.

P1 = (X1,Y1,Z1),P2=(X2,Y2,Z2) = x,y,z coordinates of the expansion PWS monopole endpoints.

A = wire radius.

D1,CGD1,SGD1 = same as DM, CGDM, SGDM defined in subroutines ZWTPE, ZWTDE and ZWTWE.

D2,SGD2 = same as DN, SGDN defined in subroutines ZWTPE, ZWTDE and ZWTWE.

THE FOLLOWING ARE OUTPUTS:

Z12 = the mutual impedance between the two piecewise sinusoidal filaments.

NOTES:

Depending on the orientation and separation of the two filaments, subroutine ZGSMM sets the integration parameter INT to either zero or two. Then it calls GGS1 which calculates the mutual impedance by a closed form expression (INT = 0) or by a two interval Simpson's rule integration (INT = 2).

19. SUBROUTINE GGS1

PURPOSE:

Subroutine GGS1 calculates the mutual impedance between two filamentary monopoles with sinusoidal current distribution. This subroutine is the same as subroutine GGS in reference [1] except that GAM = jk where k is real.

GENERAL FORM

```
GGS1(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AM,DS,CGDS,SGDS,DT,SGDT,
INT,ETA,GAM,P11,P12,P21,P22) .
```

THE FOLLOWING PARAMETERS ARE INPUTS:

PA = (XA,YA,ZA), PB=(XB,YB,ZB) = x,y,z coordinates of the test monopole endpoints.

P1 = (X1,Y1,Z1), P2=(X2,Y2,Z2) = x,y,z coordinates of the expansion filament endpoints.

AM = wire radius.

DS = length of the test filament.

SGDS = sinh(GAM*DS) .

CGDS = cosh(GAM*DS) .

DT = length of the expansion filament.

SGDT = sinh(GAM*DT) .

INT = the number of Simpson's rule intervals in the integration over the expansion filament. If INT = 0, the integration is done in closed form. Otherwise, INT is always a even integer number.

ETA = complex impedance of free space (376.7 + j0.0).

GAM = complex propagation constant of free space (0.0 - jk)

where k = 2*PI/WV.

THE FOLLOWING ARE OUTPUTS:

P11, P12, P21 and P22 = mutual impedance between the two filaments. The first subscript refers to the endpoint of filament one with non zero current. The second subscript refers to the endpoint of filament two with non-zero current.

20. SUBROUTINE DSKTS2

PURPOSE:

Subroutine DSKTS2 calculates the mutual impedance between a disk test monopole M and a wire, polygonal surface patch or disk expansion monopole N. The current distributions on the wire, polygonal surface patch and disk monopole are given by Equations (2.11), (2.13) and (2.16), respectively.

GENERAL FORM:

```
DSKTS2(XMO,YMO,ZMO,XM1,YM1,ZM1,XM2,YM2,ZM2,JOP,XN1,YN1,ZN1,XN2,YN2,ZN2,
XN3,YN3,ZN3,XN4,YN4,ZN4,IN12,INTP,NDT,BM,BN,ZMN) .
```

THE FOLLOWING ARE INPUTS:

```
COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .
```

PO = (XMO,YMO,ZMO) = x,y,z coordinates of the disk monopole center.

PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2) = x,y,z coordinates of two points on the disk plane.

JOP = expansion monopole type indicator.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3),

PN4 = (XN4,YN4,ZN4) = x,y,z coordinates of four points on the expansion monopole. If JOP = 1, PN1,PN2,PN3,PN4 are the four corners of the plate. If JOP = 2, PN1,PN2,PN3 are the same as PO,PM1,PM2 for the test disk monopole and PN4 is not used. If JOP = 3, PN1,PN2 are the wire endpoints and PN3,PN4 are not used.

IN12 = polarity indicator for the direction of current flow on the expansion monopole N.

INTP = the number of Simpson's rule intervals in the integration over the expansion monopole used in subroutines ZWTPE2 (if JOP = 1) or ZWTDE (if JOP = 2).

NDT = the number of Simpson's rule integration intervals used in the integration over the disk test monopole.

BM = outer test disk monopole radius.

BN = outer expansion disk monopole radius (If JOP = 2).

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between test monopole M and expansion monopole N.

NOTES:

The disk test monopole is represented by NDT filaments and the impedance of each filament to the expansion monopole is evaluated by calling the appropriate subroutine; i.e., ZWTPE2 for a surface patch expansion monopole, ZWTDE for a disk expansion monopole and ZWTWE for a wire expansion monopole. All those partial impedances are summed using a Simpson's rule weighting to give ZMN.

21. SUBROUTINE DSKTST

PURPOSE:

Subroutine DSKTST calculates the mutual impedance between a disk test monopole M and a wire, rectangular surface patch or disk expansion monopole N. The current distributions on the wire, surface patch and disk monopoles are given by Equations (2.11), (2.12) and (2.16) respectively.

GENERAL FORM:

```
DSKTST(XMO,YMO,ZMO,XM1,YM1,ZM1,XM2,YM2,ZM2,JOP,XN1,YN1,ZN1,XN2,YN2,
ZN2,XN3,YN3,ZN3,IN12,INTD,NINT,BM,BN,ZMN) .
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

PO = (XMO,YMO,ZMO) = x,y,z coordinates of the disk monopole center.

PM1 = (XM1,YM1,ZM1),PM2=(XM2,YM2,ZM2) = x,y,z coordinates of two points on the plane of the disk.

JOP = expansion monopole type indicator.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3) = x,y,z coordinates of three points on the expansion monopole. If N is a plate, PN1,PN2,PN3 are three consecutive corners; if N is a disk, PN1,PN2,PN3 are the same as PO,PM1, PM2 for a test disk monopole; if N is a wire, PN1,PN2 are its endpoints and PN3 is not used.

IN12 = polarity indicator for the direction of current flow on expansion monopole N.

INTD = number of Simpson's rule intervals used in integrating over the disk test monopole.

NINT = number of Simpson's rule intervals in the integration over the expansion surface patch monopole (if JOP =1) or the expansion disk monopole (if JOP = 2). The integrations are implemented in subroutines ZWTPE (if JOP = 1) or ZWTDE (if JOP = 2).

BM = outer radius of test disk monopole.

BN = outer radius of expansion disk monopole (if JOP = 2).

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between monopole M and monopole N.

NOTES:

A disk test monopole is represented by INTD filaments and the mutual impedance between every filament and the expansion monopole is calculated using ZWTPE, ZWTDE, ZWTWE depending on the type of expansion monopole. Then all those partial impedances are summed using a Simpson's rule weighting.

22. SUBROUTINE ZATAT2

PURPOSE:

Subroutine ZATAT2 calculates the self impedance of an attachment mode.

GENERAL FORM:

ZATAT2(B,H,Z,NL,ZS,ALFD) .

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

B = outer disk radius.

H = the length of wire part of attachment mode.

NL = number of intervals for trapezoidal rule integration over the disk monopole.

ZS = complex wire surface impedance.

ALFD = the angle between the axis of the wire monopole of the attachment mode and the line perpendicular to the disk plane. If ALFD = 0.0, the wire is perpendicular to the disk.

THE FOLLOWING ARE OUTPUTS:

Z = the self impedance of the attachment mode.

NOTES:

The self impedance of an attachment with a perfectly conducting wire is the sum of four partial impedances, i.e.,

$$Z = Z_{dd} + Z_{dw} + Z_{wd} + Z_{ww}$$

where Z_{dd} is the disk/disk impedance, Z_{dw} is the disk/wire impedance, Z_{wd} is the wire/disk impedance and Z_{ww} is the wire/wire impedance. Those partial impedances are evaluated by a trapezoidal rule of integration using surface testing and expansion filaments. Each filament-to-filament impedance is evaluated by GGMM1 or GGS1.

23. SUBROUTINE PDPZ1

PURPOSE:

Subroutine PDPZ1 calculates the mutual impedance between a disk monopole parallel to a non-rectangular surface patch monopole.

GENERAL FORM:

```
PDPZ1(XMO,YMO,ZMO,K,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,YN4,
ZN4,IACN,IN12,INTP,ERVSR,IAT,RMINK,DRK, ZMN,DIST) .
```

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A,Q,GAM,ETA,XK .

PO = (XMO,YMO,ZMO) = x,y,z coordinates of the disk monopole center.

K = wire attachment mode 1.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3), PN4=(XN4,YN4,ZN4) = x,y,z coordinates the four expansion surface patch monopole corners.

IACN = monopole shape indicator for identifying the type of surface patch monopoles of the expansion mode.

IN12 = polarity indicator for the direction of current flow on the surface patch expansion monopole.

INTP = the number of Simpson's rule integration intervals used in integrating over the expansion surface patch monopole.

ERVSR(K,JJ) = array containing values of the radial component of the electric field of disk monopole K versus the radial distance ρ.

IAT = dimension indicator for the maximum number of wire attachments used in array ERVSR(IAT,400).

RMINK = the minimum value of ρ corresponding to ERVSR(K,1).

DRK = the increment in the value of ρ, i.e., ρ = RMINK + DRK*JJ.

DIST = the distance between the disk monopole and surface patch monopole planes.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the disk monopole and the surface patch monopole.

NOTES:

Subroutine PDPZ1 uses arrray ERVSR to interpolate the value of the ρ component of the disk monopole electric field E and then evaluates ZMN = $\int \underline{E} \cdot \underline{J}$, where J is the current density on the expansion surface patch monopole, by an NPE interval double Simpson's rule integration.

24. SUBROUTINE PDPZ

PURPOSE:

Subroutine PDPZ calculates the mutual impedance between a disk monopole parallel to a rectangular surface patch monopole. The current density $\underline{J}n$ on the rectangular surface patch monopole is given by Equation (2.12).

GENERAL FORM:

PDPZ(XMO,YMO,ZMO,K,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,
ERVSR,IAT,RMIN,DR,ZMN,DIST) .

THE FOLLOWING ARE INPUTS:

COMMON /A/ WV,PI,A Q,GAM,ETA,XK .

PO = (XMO,YMO,ZMO)= x,y,z coordinates of disk monopole center.

K = wire attachment mode K.

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3) = x,y,z coordinates of three consecutive corners of the expansion rectangular patch monopole.

IN12 = polarity indicator for the direction ꟾf current flow on the expansion monopole.

INTP = number of Simpson's rule integration intervals used in integrating over the surface patch monopole. This integration is implemented in the DO 10 loop.

ERVSR(K,JJ) = array that contains values of the radial component of the electric field of disk monopole k versus the radial distance ρ.

IAT = dimension indicator for maximum number of attachments, used in array ERVSR(IAT,400).

RMIN) = the minimum value of ρ, corresponding to ERVSR(NAT,1).

DR = the increment in the value of ρ, i.e., ρ = RMIN + DR*JJ.

DIST = distance between the planes of the disk monopole and the rectangular surface patch monopole.

THE FOLLOWING ARE OUTPUTS:

ZMN = the mutual impedance between the disk monopole and the rectangular surface patch monopole.

NOTES:

Subroutine PDPZ uses array ERVSR to interpolate the values of the ρ component of the disk test monopole electric field Em. Then ZMN, which is $\int \underline{E}m\underline{J}n$, is evaluated by an INTP interval Simpson's rule integration.

25. SURBROUTINE ERDSK

PURPOSE:

Subroutine ERDSK calculates the near zone electric field of a disk monopole in the x-y plane with a current density given by Equation (2.16). The electric field has a ρ and z component.

GENERAL FORM:

ERDSK(A,B,X,Z,ETA,WV,NNPTS,EX) .

THE FOLLOWING ARE INPUTS:

A = inner radius of disk monopole.

B = outer radius of disk monopole.

X = the radial distance from the origin (ρ) where $E\rho$ is to be evaluated.

Z = the z coordinate of the field observation point.

ETA = complex impedance of free space.

WV = wavelength.

NNPTS = number of Simpson's rule intervals used in the ϕ integration over the disk monopole.

THE FOLLOWING ARE OUTPUTS:

EX = the value of the ρ component of the disk monopole electric field in ohms/meter.

NOTES:

Subroutine ERDSK represents the disk monopole by NNPTS radial filaments. The near zone field of every filament is calculated by using its closed form expression. This formula is described in reference [1]. The total near zone field is the Simpson's rule weighted sum of all the filamentary near zone fields.

26. SUBROUTINE COUPLE

PURPOSE:

Subroutine COUPLE finds the coupling between two ports of the wire structure of the antenna.

GENERAL FORM

COUPLE(ZT,ZTF,M1,M2,SN1,SN2,I12,V,NT,IFIL) .

The following parameters are inputs:

ZT(K) = complex impedance array used when IFIL = 0.

ZTF(M,N) = complex impedance matrix used when IFIL = 1.

M1 = wire segment "location" of first feed port.

M2 = wire segment "location" of second feed port.

SN1 = sign factor for first feed port. If port 1 is not by the terminal point of a wire dipole mode, then SN1 = 1.0. Otherwise SN1 = -1.0.

SN2 = sign factor for second feed port. If port 1 is not by the terminal point of a wire dipole mode, then SN2 = 1.0. Otherwise SN2 = -1.0.

I12 = matrix inversion indicator. SQROT or CROUT solves the matrix equation [Z][I]=[V]. In doing so, [Z] is transformed to an effective inverse. If I12 = 2, this indicates that [Z] is already the effective inverse. If I12 = 1, this indicates that [Z] is the original matrix.

[V] = dummy column vector.

NT = the total number of modes.

IFIL = indicator for choosing between full surface patch test monopoles (IFIL = 0) or filamentary surface patch test monopoles (IFIL = 1).

NOTES:

Let ZT or ZTF be the impedance matrix of the problem and [Z][I] = [V] or [Y][V] = [I] where Z = ZT or ZTF. If we set the V column vector to zero except for the entries corresponding to the two ports of interest, we can reduce the [Y][V] = [I] matrix equation to

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

where the subscripts 1,2 do not represent actual locations in the original Y matrix but ports 1 and 2. The Y11, Y12, Y21 and Y22 are determined as follows:

Let the [V] column vector of the original matrix be equal to zero except for the entry corresponding to port 1 which is set to (1.0,0.0). Then the [V] column vector is the input to SQROT or CROUT and comes out as the "induced" current vector [V]; Y11 = V(M1) and Y12 = V(M2). To find the other two parameters the same procedure is used except that the entry corresponding to port 2 is set to (1.0,0.0). Again Y21 = V(M1) and Y22 = V(M2).

Finally, the admittance matrix is inverted to obtain the impedance matrix Z relating the two ports. COUPLE also finds the maximum coupling between the two ports.

27. SUBROUTINE ANTV

PURPOSE:

Subroutine ANTV evaluates the current vector I of the matrix equation [Z][I]= [V] (see Equation 2.8) for an antenna problem, i.e., when the excitation is due to a delta gap generator. It also calculates the input impedance of the antenna, the power dissipated by the wire structure and the efficiency of the antenna.

GENERAL FORM:

ANTV(I1,I2,I3,IA,IB,IWR,JA,JB,NM,ZT,IFIL,ICC,ZTF,CJ,VG,Y11,Z11,NWR,
NPL,NAT,VGA,PIN,AM,CMM,D,DISS,GAM,SGD,ZLD,ZS,ZLDA,INM,MD,ND,NSA)

THE FOLLOWING ARE INPUTS:

I1(I) = endpoint 1 of wire dipole I.

I2(I) = terminal point of wire dipole I.

I3(I) = endpoint 2 of wire dipole I.

IA(J) = endpoint A of wire segment J.

IB(J) = endpoint B of wire segment J.

IWR = write indicator used by subroutine SQROT. If IWR = 1, SQROT writes out the modal currents.

JA(I) = segment number of the first segment of wire dipole I.

JB(I) = segment number of the second segment of wire dipole I.

NM = total number of wire segments.

ZT(K) = complex impedance array used when IFIL = 0.

IFIL = indicator for choosing either full surface patch test modes (IFIL = 0) or filamentary surface patch test modes (IFIL = 1).

ICC = dimension indicator for array ZTF(ICC,ICC).

ZTF(M,N) = complex impedance matrix used when IFIL = 1.

VG(II) = complex voltage generator at wire "location" II.

NWR = total number of wire modes.

NPL = total number of plate modes.

NAT = total number of wire attachment dipole modes.

VGA(K) = complex voltage generator at wire attachment K.

AM = wire radius.

CMM = wire conductivity in megamhos/meter. CMM = -1.0 implies a perfect conductor.

D(J) = length of wire segment J.

GAM = complex free space propagation constant defined by Equation (4.1).

SGD(J) = sinh(GAM*D(J))

ZLD(II) = complex impedance loading at wire "location" II.

ZS = complex surface impedance of the wire.

ZLDA(K) = complex impedance loading at wire attachment K.

INM = dimension indicator for array MD(INM,4).

MD(J,L) = list of wire dipoles sharing wire segment J.

ND(J) = total number of wire dipoles sharing wire segment J.

NSA(K) = wire segment "location" of attachment K.

THE FOLLOWING ARE OUTPUTS:

CJ(I) = magnitude of current of mode I.

CG(I) = magnitude of current at wire segment "location" I.

PIN = time average power input to the wire structure.

DISS = time average power dissipated by the wire structure.

Y11 = complex input admittance(mohms).

Z11 = complex input impedance(ohms).

NOTES:

Consider the matrix equation [Z][I]=[V]. ANTV finds all the entries CJ of the V column and stores them in array CG before it calls SQROT or CROUT. Upon entry to SQROT or CROUT, CJ is the excitation column V. On exit, the solution vector I is stored in array CJ.

In the DO 80 loop, ANTV calculates the admittance Y11 of the antenna using the formula Y11 = Current X Conjugate(voltage). Actually Y11 is the input admittance only if the antenna is fed by a single one-volt generator.

Finally, ANTV calls AGDISS to evaluate the power dissipated in antenna structure and ARITE to find the branch currents, i.e., the currents at each wire "location".

28. SUBROUTINE AGDISS

PURPOSE:

Subroutine AGDISS calculates the time average power dissipated by the imperfectly conducting wire structure of the antenna and its loads.

GENERAL FORM:

AGDISS(AM,CG,CMM,D,DISS,GAM,NM,SGD,ZLD,ZS,ZLDA,NAT,NSA)

THE FOLLOWING ARE INPUTS:

AM = wire radius.

CG(I) = magnitude of current at wire "location" I.

CMM = wire conductivity in megamhoms/meter. If CMM = -1.0, the wire is a perfect conductor.

D(I) = length of wire mode I.

GAM = complex free space propagation constant.

NM = total number of wire segments.

SGD = sinh(GAM*D)

ZLD(I) = complex impedance load at wire "location" I.

ZS = complex wire surface impedance.

ZLDA(K) = complex impedance load at wire attachment K.

NAT = total number of wire attachments.

NSA(K) = wire "location" of attachment K.

THE FOLLOWING ARE OUTPUTS:

NOTES:

DISS = the time average power dissipated in the wire and the loads.

Subroutine AGDISS uses Poynting's theorem to calculate the time average power dissipated in the imperfect wire in the DO 100 loop. The power dissipated in the loads is calculated in the DO 140 loop using the formula Power = Impedance^{x}(Current)^{2}.

29. SUBROUTINE ARITE

PURPOSE:

Subroutine ARITE generates a list of branch currents using the loop currents.

GENERAL FORM:

ARITE(IA,IB,INM,IWR,I1,I2,I3,MD,ND,NM,CJ,CG,NSA,NWR,NPLTM,NAT).

THE FOLLOWING ARE INPUTS:

IA(J) = endpoint A of wire segment J.

IB(J) = endpoint B of wire segment J.

INM = dimension indicator for array MD(INM,4).

IWR = indicator for writing out the induced modal currents.

I1(J) = endpoint 1 of wire dipole mode J.

I2(J) = terminal point of wire dipole mode J.

I3(J) = endpoint 2 of wire dipole mode J.

MD(J,L) = list of wire dipoles sharing wire segment J.

ND(J) = total number of wire dipoles sharing wire segment J.

NM = total number of wire segments.

CJ(I) = amplitude of the current of mode I.

NSA(K) = wire "location" of attachment K.

NWR = total number of wire modes.

NPLTM = total number of plate modes.

NAT = total number of wire attachments.

THE FOLLOWING ARE OUTPUTS:

CG(J) = amplitude of the current at wire "location" J.

30. SUBROUTINE CROUT

PURPOSE:

Consider the matrix equation [Z][I] =[V] which represents a system of simultaneous linear equations. Given the excitation vector V and the matrix Z subroutine, CROUT solves for vector I.

GENERAL FORM:

CROUT(S,ICC,C,ISYM,IWR,I12,N)

THE FOLLOWING ARE INPUTS:

S(I) = complex array containing the excitation vector V.

ICC = dimension indicator for array C(I,J).

C(I,J) = complex matrix.

ISYM = symmetry indicator. If ISYM = 0, array C(I,J) is not symmetric and if ISYM = 1, array C(I,J) is symmetric.

IWR = write indicator for printing the solution vector.

I12 = matrix inversion indicator. CROUT solves the matrix equation [Z][I]=[V]. In doing so, [Z] is transformed to an effective inverse. If I12 = 2, this indicates that [Z] is already the effective inverse. If I12 = 1, this indicates that [Z] is the original matrix.

THE FOLLOWING ARE OUTPUTS:

S(K) = complex array containing the solution vector I.

C(I,J) = effective inverse of input matrix C(I,J).

31. SUBROUTINE SORTB

PURPOSE:

Subroutine SORTB calculates the far zone electric field for an antenna problem or the backscattered electric field and various cross sections for a scattering problem.

GENERAL FORM:

```
SORTB(IA,IB,I1,I2,I3,NWR,NM,A,CGD,SGD,FHZ,D,IWRSQ,I12,ISCAT,ZTF,ZT,
IFIL,ICC,X,Y,Z,NPL,NAT,PA,PB,NSA,NPLA,PCN,BDSK,IQUAD,
NPLTM,IPL,IPLM.CJP,CJT,ETTS,EPPS,ETPS,EPTS,THETA,PHI,JA,JB,
SCSP,SCST,SPPM,SPTM,STPM,STTM,IMAGE,ICN,NDNPLT) .
```

THE FOLLOWING ARE INPUTS:

IA(I) = endpoint A of wire segment I.

IB(I) = endpoint B of wire segment I.

I1(J) = endpoint 1 of wire dipole mode J.

I2(J) = terminal point of wire dipole mode J.

I3(J) = endpoint 2 of wire dipole mode J.

NWR = total number of wire modes.

NM = total number of wire segments.

A = wire radius.

D(J) = length of wire segment J.

CGD(J) = cosh(GAM*(D(J))

SGD(J) = sinh(GAM*(D(J))

FHZ = frequency in Hertz.

IWRSQ = write indicator used in SQROT(CROUT). If IWRSQ = 1, then SQROT(CROUT) writes out the induced modal currents.

I12 = matrix ZT or ZTF inversion indicator. SQROT or CROUT solves the matrix equation [Z][I]=[V]. In doing so, [Z] is transformed to an effective inverse. If I12 = 2, this indicates that [Z] is already the effective inverse. If I12 = 1, this indicates that [Z] is the original matrix.

ISCAT = indicator for choosing between antenna calculations (ISCAT = 0), backscattering (ICAT = 1) and bistatic scattering (ISCAT = 2).

ZTF(M,N)= complex impedance matrix used when IFIL = 1.

ZT(K) = complex impedance array used when IFIL = 0.

IFIL = indicator for choosing either full surface patch test modes (IFIL = 0) or filamentary surface patch plate modes (IFIL = 1).

ICC = dimension indicator for matrix ZTF(ICC,ICC).

X(I),Y(I),Z(I) = x,y,z coordinates of point I of the antenna wire structure.

NPL = total number of plates.

NAT = total number of attachments.

PA(I,J,K) = x,y,z cordinates (K=1,2,3) of the J-th corner of monopole A of the I-th surface patch dipole mode.

PB(I,J,K) = x,y,z coordinates (K=1,2,3) of the J-th corner of monopole B of the I-th surface patch dipole mode.

NSA(K) = the wire segment "location" of attachment K.

NPLA(K) = number of plate where wire attachment K is located.

PCN(I,J,K) = x,y,z coordinates (I=1,2,3) of the J-th corner of the K-th plate.

BDSK(K) = outer radius of disk monopole of attachment dipole mode K.

IQUAD(I) = monopole shape indicator for identifying the type of monopoles of plate mode I.

IQUAD(I) = -3 implies both monopoles of plate dipole mode I are rectangular.

IQUAD(I) = 0 implies that either or both monopoles of plate dipole mode I are quadrilateral surface patches.

NPLTM = total number of plate modes.

IPL = dimension indicator for the maximum number of plates used in array PCN(3,ICN,IPL).

IPLM = dimension indicator for the maximum number of plate modes used in arrays PA(3,4,IPLM) and PB(3,4,IPLM).

CJP(I) = array containing the magnitude of the induced modal current on mode I due to a ϕ polarized incident wave. This array is an input from subroutine ANTV only when ISCAT = 0.

CJT(I) = same as CJP(I) but for θ polarization.

THETA = elevation angle of observation point.

PHI = azimuth angle of observation point.

JA(I) = segment number of the first segment of wire dipole mode I.

JB(I) = segment number of the second segment of wire dipole mode I.

IMAGE = incident wave image indicator.

IMAGE = 0 implies no image incident wave is desired.

IMAGE = 1 implies the image of the incident wave is desired. The image plane is the x-y plane. The program automatically includes the image of the incident wave when IMAGE = 1 but the user has to define the image geometry.

ICN = dimension indicator for the maximum number of plate corners used in array PCN(3,ICN,IPL).

NDNPLT(I) = total number of plate modes through plate I.

THE FOLLOWING ARE OUTPUTS:

When ISCAT = 0 (radiation problem) the only outputs are:

ETTS = θ component of far zone radiated electric field.

EPPS = ϕ component of far zone radiated electric field.

When ISCAT = 1 or 2 (scattering or backscattering problem) the outputs are:

ETTS = θ component of far zone scattered electric field due to a θ polarized incident wave.

EPPS = ϕ component of far zone scattered electric field due to a ϕ polarized incident wave.

ETPS = θ component of far zone scattered electric field due to a ϕ polarized incident wave.

EPTS = ϕ component of far zone scattered electric field due a θ polarized incident wave.

ECSP = extinction cross section due to a ϕ polarized incident wave.

ECST = extinction cross section due to a θ polarized incident wave.

SCSP = scattering cross section due to a ϕ polarized incident wave.

SCST = scattering cross section due to a θ polarized incident wave.

SPPM = echo area of ϕ polarized scattered electric field due to a ϕ polarized incident wave.

SPTM = echo area of ϕ polarized scattered electric field due to a θ polarized incident wave.

STPM = echo area of θ polarized scattered electric field due to a ϕ polarized incident wave.

STTM = echo area of θ polarized scattered electric field due to a θ polarized incident wave.

CJP(I) = array containing the magnitude of the induced current on mode I for a ϕ polarized incident wave. It is an output only when ISCAT = 1 or 2.

CJT(I) = array containing the magnitude of the induced current on mode I for a θ polarized incident wave. It is an output only when ISCAT = 1 or 2.

NOTES:

For ISCAT = 0, subroutine SORTB calculates the far zone electric field radiated by the antenna structure. This is done by a superposition of the far zone electric fields of all the expansion modes. The field of a particular mode is obtained by calling the appropriate far zone subroutine (GFF for a wire expansion, SURMFF for a surface patch, DSKFF for a disk monopole of an attachment mode) which outputs the E field as ET (E_θ) and EP (E_ϕ). ET and EP are weighted by the corresponding loop current CJP, which is evaluated in ANTV, to give the actual fields ETTS (E_θ) and EPPS (E_ϕ) of that particular mode. All this is done within the DO 160 loop.

For ISCAT = 1, subroutine SORTB calculates the backscattered far zone electric field from the scatterer. This is done by a superposition of the backscatttered electric field of all expansion modes. The backscattered field of a particular mode is evaluated as follows: the ET and EP components of the mode are calculated as for ISCAT = 0 and stored in the arrays ETT and EPP, respectively. The excitation voltages arrays CJP and CJT, described by Equation (10-1), are obtained by multiplying CJI with EP and ET, respectively, and are used by SQROT to obtain the induced loop current arrays CJP and CJT. (Arrays CJP and CJT contain the excitation voltages upon entry to SQROT(CROUT) and upon exit from SQROT(CROUT) they contain the induced loop currents). Finally, ETT and EPP are weighted by CJT and CJP to give the actual backscattered E field of the mode.

An ISCAT = 2 is the same as ISCAT =1 except that the subroutine does not have to evaluate the excitation voltage arrays CJT and CJP. These are calculated by a preceding call to SORTB with ISCAT = 1.

32. SUBROUTINE SURMFF

PURPOSE:

Subroutine SURMFF calculates the far zone electric field of a rectangular surface patch monopole in the y-z plane. The current density on the monopole is given by Equation (2.12).

GENERAL FORM:

SURMFF(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,I12,TH,PH,ETH,EPH,WVL) .

THE FOLLOWING ARE INPUTS:

P1 = (X1,Y1,Z1),P2=(X2,Y2,Z2),P3=(X3,Y3,Z3) = x,y,z coordinates of three consecutive corners of the rectangular surface patch monopole.

I12 = polarity indicator for the direction of current flow on the expansion surface patch monopole.

TH = elevation angle of observation point.

PH = azimuth angle of observation point.

WVL = wavelength in meters.

THE FOLLOWING ARE OUTPUTS:

ETH = θ component of far-zone electric field E.

EPH = ϕ component of far-zone electric field E.

NOTES:

This subroutine is an implementat'on of Equation (103) of [10].

33. SUBROUTINE SURFFP

PURPOSE:

Subroutine SURFFP calculates the far zone electric field electric field of a polygonal surface patch monopole in the y-z plane. The current density on the monopole is given by Equation (2.13).

GENERAL FORM:

```
SURFFP(XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,YN4,ZN4,NPLS,GAM,ETA,XK,
FHZ,THR,PHR,IN12,ETH,EPH) .
```

THE FOLLOWING ARE INPUTS:

PN1 = (XN1,YN1,ZN1),PN2=(XN2,YN2,ZN2),PN3=(XN3,YN3,ZN3),
PN4=(XN4,YN4,ZN4) = x,y,z coordinates of the four corners of the quadrilateral surface patch monopole.

NPLS = number of Simpson's rule integration intervals used in integrating over the surface patch monopole.

GAM = complex propagation constant of free space.

ETA = complex impedance of free space.

XK = k, the free space wave number.

FHZ = frequency in Hertz.

THR = elevation angle of observation point.

PHR = azimuth angle of observation point.

IN12 = polarity indicator for the direction of current flow on the surface patch monopole.

THE FOLLOWING ARE OUTPUTS:

ETH = θ component of far-zone electric field E.

EPH = ϕ component of far-zone electric field E.

NOTES:

Subroutine SURFFP represents the polygonal surface patch monopole with NPLS filaments and calls subroutine GFF to evaluate the far zone electric field of each filament. Then it sums all those filamentary fields using a Simpson's rule weighting to obtain the total far zone electric field of the surface patch monopole.

34. SUBROUTINE DSKFF

PURPOSE:

Subroutine DSKFF calculates the far zone electric field of a disk monopole located in the x-y plane. The surface current density is given by Equation (2.16).

GENERAL FORM:

DSKFF(XO,YO,ZO,X1,Y1,Z1,X2,Y2,Z2,TH,PH,A,B,WVL,ETH,EPH) .

THE FOLLOWING ARE INPUTS:

PO = (XO,YO,ZO) = x,y,z coordinates of disk monopole center.

P1 = (X1,Y1,Z1),P2=(X2,Y2,Z2) = x,y,z coordinates of two points on the disk monopole plane.

TH = elevation angle of the observation point.

PH = azimuth anqle of the observation point.

A = inner radius of the disk monopole.

B = outer radius of the disk monopole.

WVL = wavelength in meters.

THE FOLLOWING ARE OUTPUTS:

ETH = θ component of far-zone electric field E.

EPH = ϕ component of far-zone electric field E.

NOTES:

DSKFF is an implementation of Equation (113) of [10].

CHAPTER V

SUMMARY

The description of a general purpose computer code based on the Moment Method (MM) solution for electromagnetic radiation and scattering problems has been presented. The MM formulation as implemented by the program was discussed in brief. The program inputs were described and several examples illustrating their use were given. Finally, all the program subroutines were described.

The program can handle geometries consisting of wire, polygonal plates, wire/plate junctions and multiple junctions. Since the computation time is proportional to the square of the number of modes, the program is limited to low frequencies or, equivalently, it is practical up to frequencies that do not make the number of modes prohibitively large. The major advantages of the program are accuracy, flexibility and the simplicity of the input format.

REFERENCES

[1] J.H. Richmond, "Computer Program for Thin-Wire Structures in a Homogeneous Conducting Medium", Report 2903-12, August 1973, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Grant NGL 36-008-138 for National Aeronautics and Space Administration (NASA-CR-2399).

[2] H.H. Chao and B.S. Strait, "Computer Programs for Radiation and Scattering by Arbitrary Configurations of Bent Wires", Scientific Report No. 7, AFCRL-70-0374, Syracuse University, Syracuse, New York, September 1970.

[3] J.H. Richmond, "A Wire-Grid Model for Scattering by Conducting Bodies", IEEE Trans. Antennas and Propagation, Vol. A-14, pp. 782-786, November 1966.

[4] F.K. Oshiro, "Source Distribution Techniques for the Solution of General Electromagnetic Scattering Problems", Proceedings First GISAT Symposium, Mitre Corp., Vol. I, pp. 83-107, 1965.

[5] N.C. Albertsen, J.E. Hansen and N.E. Hansen, "Computation of Radiation for Wire Antennas on Conducting Bodies", IEEE Trans. Antennas and Propagation, Vol. AP-22, pp. 200-206, March 1974.

[6] G.J. Burke and A.J. Poggio, "Numerical Electromagnetic Code-Method of Moments", Naval Ocean Systems Center technical document 116, AFWL-TR-76 [illegible], July 1977.

[7] N.N. Wang, J.H. Richmond and M.C. Gilreath, "Sinusoidal Reaction Formulation for Radiation and Scattering from Conducting Surfaces", IEEE Trans. Antennas and Propagation, Vol. AP-23, pp. 376-382, May 1975.

[8] D.R. Wilton, S.S.M. Rao and A.W. Glisson, "Electromagnetic Scattering by Surfaces of Arbitrary Shape", Interaction Notes, Note 388, Department of Electrical Engineering, University of Mississippi, September 1979.

[9] E.H. Newman and D.M. Pozar, "Electromagnetic Modeling of Composite Wire and Surface Geometries", IEEE Trans. Antennas and Propagation, Vol. AP-26, pp. 784-789, November 1978.

[10] D.M. Pozar, "On Moment Method Solutions for Plate and Wire Geometries", Ph.D. dissertation, Ohio State University, 1980.

[11] Pravit Tulyathan, "Moment Method Solutions for Radiation and Scattering from Arbitrarily Shaped Surfaces", Ph.D. dissertation, Ohio State University, 1981.

[12] V.H. Rumsey, "Reaction Concept in Electromagnetic Theory", Physical Review, Vol. 94, pp. 1483-1491, June 15, 1954.

[13] S.A. Schelkunoff, "On Diffraction and Radiation of Electromagnetic Waves", Physical Review, Vol. 56, August 15, 1939.

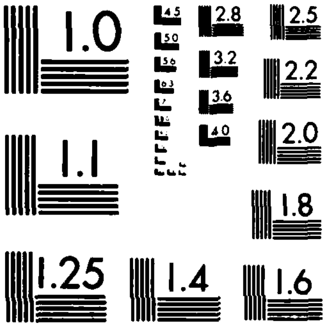
[14] E.H. Newman and M.R. Schrote, "On the Current Distribution for Open Surfaces", IEEE Trans. Antennas and Propagation, Vol. AP-31, pp. 515-518, May 1983.

[15] S.A. Schelkuroff and H.T. Friis, Antennas: Theory and Practice, John Wiley and Sons, New York, 1952.

[16] J.H. Richmond, D.M. Pozar and E.H. Newman, "Rigorous Near-Zone Field Expressions for Rectangular Sinusoidal Monopole", IEEE Trans. Antennas and Propagation, AP-26, p. 509, May 1978.

[17] J.H. Richmond and N.H. Geary, "Mutual Impedance of Nonplanar-Skew Sinusoidal Dipoles", IEEE Trans. Antennas and Propagation, Vol. AP-23, May 1975.

AD-A135 837   A USER'S MANUAL FOR ELECTROMAGNETIC SURFACE PATCH (ESP)   3/4
CODE VERSION II P..(U) OHIO STATE UNIV COLUMBUS
ELECTROSCIENCE LAB E H NEWMAN ET AL. SEP 83
UNCLASSIFIED   ESL-712692-3 N00014-78-C-0049   F/G 20/14   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

APPENDIX 1

OUTPUT FOR DESIGN EXAMPLE 1

```
        INPUT DATA

FREQ.(MHZ) =   150.000   WAVE(M) =     2.000   WIRE RADIUS(M) =  0.0010000
INTP=  10   INTD=  18   INT =    4   IFIL= 1

WIRE CONDUCTIVITY = 38.00 MEGAMHOS/M


GEOMETRY FOR THE   1 PLATES


                         PLATE NUMBER   1(RECTANGULAR)
NUMBER OF CORNERS =   4
MAXIMUM SEGMENT SIZE  (METERS) =    0.12500
POLARIZATION INDICATOR =   3
GENERATING SIDE INDICATOR =   0

X,Y,Z COOR.(METERS) OF CORNER  1 =    -0.50000    -0.50000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  2 =     0.50000    -0.50000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  3 =     0.50000     0.50000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  4 =    -0.50000     0.50000     0.00000


COOR.(METERS) OF  24 MODES ON THIS PLATE

MONOPOLE  MODE    X1       Y1       Z1       X2       Y2       Z2       X3       Y3       Z3       X4       Y4       Z4

    A      1    0.50000 -0.25000  0.00000  0.50000 -0.50000  0.00000  0.25000 -0.50000  0.00000  0.25000 -0.25000  0.00000
    B      1    0.50000 -0.25000  0.00000  0.50000  0.00000  0.00000  0.25000  0.00000  0.00000  0.25000 -0.25000  0.00000

    A      2    0.50000  0.00000  0.00000  0.50000 -0.25000  0.00000  0.25000 -0.25000  0.00000  0.25000  0.00000  0.00000
    B      2    0.50000  0.00000  0.00000  0.50000  0.25000  0.00000  0.25000  0.25000  0.00000  0.25000  0.00000  0.00000

    A      3    0.50000  0.25000  0.00000  0.50000  0.00000  0.00000  0.25000  0.00000  0.00000  0.25000  0.25000  0.00000
    B      3    0.50000  0.25000  0.00000  0.50000  0.50000  0.00000  0.25000  0.50000  0.00000  0.25000  0.25000  0.00000

    A      4    0.25000 -0.25000  0.00000  0.25000 -0.50000  0.00000  0.00000 -0.50000  0.00000  0.00000 -0.25000  0.00000
    B      4    0.25000 -0.25000  0.00000  0.25000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000 -0.25000  0.00000

    A      5    0.25000  0.00000  0.00000  0.25000 -0.25000  0.00000  0.00000 -0.25000  0.00000  0.00000  0.00000  0.00000
    B      5    0.25000  0.00000  0.00000  0.25000  0.25000  0.00000  0.00000  0.25000  0.00000  0.00000  0.00000  0.00000

    A      6    0.25000  0.25000  0.00000  0.25000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.25000  0.00000
    B      6    0.25000  0.25000  0.00000  0.25000  0.50000  0.00000  0.00000  0.50000  0.00000  0.00000  0.25000  0.00000
```

```
A       7    0.00000 -0.25000  0.00000  0.00000 -0.50000  0.00000 -0.25000 -0.50000  0.00000 -0.25000 -0.25000  0.00000
B       7    0.00000 -0.25000  0.00000  0.00000  0.00000  0.00000 -0.25000  0.00000  0.00000 -0.25000 -0.25000  0.00000

A       8    0.00000  0.00000  0.00000  0.00000 -0.25000  0.00000 -0.25000 -0.25000  0.00000 -0.25000  0.00000  0.00000
B       8    0.00000  0.00000  0.00000  0.00000  0.25000  0.00000 -0.25000  0.25000  0.00000 -0.25000  0.00000  0.00000

A       9    0.00000  0.25000  0.00000  0.00000  0.00000  0.00000 -0.25000  0.00000  0.00000 -0.25000  0.25000  0.00000
B       9    0.00000  0.25000  0.00000  0.00000  0.50000  0.00000 -0.25000  0.50000  0.00000 -0.25000  0.25000  0.00000

A      10   -0.25000 -0.25000  0.00000 -0.25000 -0.50000  0.00000 -0.50000 -0.50000  0.00000 -0.50000 -0.25000  0.00000
B      10   -0.25000 -0.25000  0.00000 -0.25000  0.00000  0.00000 -0.50000  0.00000  0.00000 -0.50000 -0.25000  0.00000

A      11   -0.25000  0.00000  0.00000 -0.25000 -0.25000  0.00000 -0.50000 -0.25000  0.00000 -0.50000  0.00000  0.00000
B      11   -0.25000  0.00000  0.00000 -0.25000  0.25000  0.00000 -0.50000  0.25000  0.00000 -0.50000  0.00000  0.00000

A      12   -0.25000  0.25000  0.00000 -0.25000  0.00000  0.00000 -0.50000  0.00000  0.00000 -0.50000  0.25000  0.00000
B      12   -0.25000  0.25000  0.00000 -0.25000  0.50000  0.00000 -0.50000  0.50000  0.00000 -0.50000  0.25000  0.00000

A      13    0.25000 -0.50000  0.00000  0.50000 -0.50000  0.00000  0.50000 -0.25000  0.00000  0.25000 -0.25000  0.00000
B      13    0.25000 -0.50000  0.00000  0.00000 -0.50000  0.00000  0.00000 -0.25000  0.00000  0.25000 -0.25000  0.00000

A      14    0.00000 -0.50000  0.00000  0.25000 -0.50000  0.00000  0.25000 -0.25000  0.00000  0.00000 -0.25000  0.00000
B      14    0.00000 -0.50000  0.00000 -0.25000 -0.50000  0.00000 -0.25000 -0.25000  0.00000  0.00000 -0.25000  0.00000

A      15   -0.25000 -0.50000  0.00000  0.00000 -0.50000  0.00000  0.00000 -0.25000  0.00000 -0.25000 -0.25000  0.00000
B      15   -0.25000 -0.50000  0.00000 -0.50000 -0.50000  0.00000 -0.50000 -0.25000  0.00000 -0.25000 -0.25000  0.00000

A      16    0.25000 -0.25000  0.00000  0.50000 -0.25000  0.00000  0.50000  0.00000  0.00000  0.25000  0.00000  0.00000
B      16    0.25000 -0.25000  0.00000  0.00000 -0.25000  0.00000  0.00000  0.00000  0.00000  0.25000  0.00000  0.00000

A      17    0.00000 -0.25000  0.00000  0.25000 -0.25000  0.00000  0.25000  0.00000  0.00000  0.00000  0.00000  0.00000
B      17    0.00000 -0.25000  0.00000 -0.25000 -0.25000  0.00000 -0.25000  0.00000  0.00000  0.00000  0.00000  0.00000

A      18   -0.25000 -0.25000  0.00000  0.00000 -0.25000  0.00000  0.00000  0.00000  0.00000 -0.25000  0.00000  0.00000
B      18   -0.25000 -0.25000  0.00000 -0.50000 -0.25000  0.00000 -0.50000  0.00000  0.00000 -0.25000  0.00000  0.00000

A      19    0.25000  0.00000  0.00000  0.50000  0.00000  0.00000  0.50000  0.25000  0.00000  0.25000  0.25000  0.00000
B      19    0.25000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.25000  0.00000  0.25000  0.25000  0.00000

A      20    0.00000  0.00000  0.00000  0.25000  0.00000  0.00000  0.25000  0.25000  0.00000  0.00000  0.25000  0.00000
B      20    0.00000  0.00000  0.00000 -0.25000  0.00000  0.00000 -0.25000  0.25000  0.00000  0.00000  0.25000  0.00000

A      21   -0.25000  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000  0.25000  0.00000 -0.25000  0.25000  0.00000
B      21   -0.25000  0.00000  0.00000 -0.50000  0.00000  0.00000 -0.50000  0.25000  0.00000 -0.25000  0.25000  0.00000

A      22    0.25000  0.25000  0.00000  0.50000  0.25000  0.00000  0.50000  0.50000  0.00000  0.25000  0.50000  0.00000
B      22    0.25000  0.25000  0.00000  0.00000  0.25000  0.00000  0.00000  0.50000  0.00000  0.25000  0.50000  0.00000

A      23    0.00000  0.25000  0.00000  0.25000  0.25000  0.00000  0.25000  0.50000  0.00000  0.00000  0.50000  0.00000
B      23    0.00000  0.25000  0.00000 -0.25000  0.25000  0.00000 -0.25000  0.50000  0.00000  0.00000  0.50000  0.00000

A      24   -0.25000  0.25000  0.00000  0.00000  0.25000  0.00000  0.00000  0.50000  0.00000 -0.25000  0.50000  0.00000
B      24   -0.25000  0.25000  0.00000 -0.50000  0.25000  0.00000 -0.50000  0.50000  0.00000 -0.25000  0.50000  0.00000
```

```
  4 POINTS ON THE WIRE
I         X (I)           Y (I)           Z (I)

1   0.0000E+00      0.0000E+00      0.0000E+00
2   0.0000E+00      0.0000E+00      0.2500E+00
3   0.0000E+00      0.0000E+00      0.5000E+00
4  -0.3000E+00      0.0000E+00      0.2500E+00


MODES ON THE WIRE STRUCTURE

MAXIMUM NUMBER OF MODES AT ONE POINT =  2
MINIMUM NUMBER OF MODES AT ONE POINT =  1
NUMBER OF WIRE MODES =   2


 I  I1(I)  I2(I)  I3(I)  JA(I)  JB(I)

1     1      2      3      1      2
2     3      2      4      2      3


  3 SEGMENTS ON THE WIRE
 J    IA(J)   IB(J)    D(J)(M)

 1     1       2     0.25000E+00
 2     2       3     0.25000E+00
 3     2       4     0.30000E+00



GEOMETRY FOR THE  1 ATTACHMENT POINTS

I  SEGMENT  END  PLATE      B(M)

1     1      0     1      0.40000

 LISTING OF LOADS AND GENERATORS

  0.5000E+02   0.0000E+00 OHMS BY PT. A OF SEGMENT   3
  0.1000E+01   0.0000E+00 VOLTS AT ATTACHMENT  1


  NWR = NUMBER OF WIRE MODES =    2
  NPLTM = NUMBER OF PLATE MODES =   24
  NAT = NUMBER OF ATTACHMENT MODES =    1


  ANTENNA MODAL CURRENTS
 MODE   REL MAG       ABS MAG      PHASE
```

```
  1     0.948     0.0106973     -71.
  2     0.515     0.0058164     -75.
  3     0.076     0.0008540     -73.
  4     0.003     0.0000329     -29.
  5     0.088     0.0009890     106.
  6     0.065     0.0007319     -72.
  7     0.001     0.0000166      46.
  8     0.051     0.0005809     105.
  9     0.094     0.0010587     -71.
 10     0.001     0.0000076       1.
 11     0.103     0.0011599     108.
 12     0.062     0.0006992     -74.
 13     0.008     0.0000849     -51.
 14     0.069     0.0007783     104.
 15     0.032     0.0003572     -52.
 16     0.022     0.0002500      64.
 17     0.080     0.0009086     100.
 18     0.080     0.0009045     -67.
 19     0.049     0.0005505      94.
 20     0.126     0.0014242     103.
 21     0.093     0.0010502     -67.
 22     0.063     0.0007057      98.
 23     0.133     0.0015004     104.
 24     0.021     0.0002376     -44.
 25     0.017     0.0001895      36.
 26     0.077     0.0008648      99.
 27     1.000     0.0112878     -66.

 INPUT ADMITTANCE(MHOS) =   0.004575 J  -0.010319
 INPUT IMPEDANCE(OHMS) =     35.905 J     80.990
 EFFICIENCY(PERCENT) =  62.481


 ANTENNA PROBLEM, ISCAT =     0


ELEVATION PATTERN. PHI =   0.0 DEG.

THETA(DEG)    GTHETA(DB)       GPHI(DB)
   0           -11.253          -38.612
   3           -10.175          -38.715
   6            -8.979          -38.644
   9            -7.885          -39.000
  12            -6.905          -39.181
  15            -6.038          -39.386
  18            -5.272          -39.613
  21            -4.599          -39.861
  24            -4.009          -40.127
  27            -3.491          -40.409
  30            -3.040          -40.705
  33            -2.647          -41.011
  36            -2.307          -41.325
  39            -2.016          -41.643
  42            -1.767          -41.961
  45            -1.557          -42.276
```

48	-1.382	-42.584
51	-1.238	-42.882
54	-1.122	-43.166
57	-1.030	-43.433
60	-0.960	-43.681
63	-0.908	-43.907
66	-0.873	-44.111
69	-0.851	-44.290
72	-0.841	-44.445
75	-0.840	-44.575
78	-0.848	-44.680
81	-0.863	-44.762
84	-0.888	-44.819
87	-0.936	-44.852
90	-0.899	-44.866
93	-0.867	-44.852
96	-0.927	-44.820
99	-0.972	-44.762
102	-1.020	-44.680
105	-1.074	-44.575
108	-1.137	-44.445
111	-1.213	-44.290
114	-1.304	-44.111
117	-1.413	-43.907
120	-1.544	-43.681
123	-1.699	-43.433
126	-1.883	-43.166
129	-2.100	-42.882
132	-2.354	-42.584
135	-2.649	-42.276
138	-2.992	-41.961
141	-3.389	-41.643
144	-3.846	-41.325
147	-4.373	-41.011
150	-4.980	-40.705
153	-5.681	-40.409
156	-6.492	-40.127
159	-7.437	-39.861
162	-8.544	-39.613
165	-9.855	-39.386
168	-11.428	-39.181
171	-13.339	-39.000
174	-15.675	-38.844
177	-18.481	-38.715
180	-18.806	-38.612
183	-17.766	-38.538
186	-15.904	-38.491
189	-13.707	-38.473
192	-11.780	-38.482
195	-10.164	-38.520
198	-8.809	-38.585
201	-7.662	-38.677
204	-6.684	-38.794
207	-5.842	-38.936
210	-5.115	-39.102
213	-4.484	-39.289
216	-3.935	-39.496
219	-3.458	-39.720
222	-3.043	-39.959
225	-2.682	-40.211

```
228        -2.370        -40.471
231        -2.101        -40.738
234        -1.870        -41.007
237        -1.672        -41.274
240        -1.504        -41.535
243        -1.363        -41.787
246        -1.245        -42.026
249        -1.149        -42.247
252        -1.071        -42.447
255        -1.011        -42.622
258        -0.967        -42.770
261        -0.938        -42.888
264        -0.926        -42.974
267        -0.943        -43.026
270        -0.899        -43.042
273        -0.868        -43.026
276        -0.924        -42.974
279        -0.977        -42.888
282        -1.042        -42.770
285        -1.123        -42.622
288        -1.222        -42.447
291        -1.342        -42.247
294        -1.486        -42.026
297        -1.657        -41.787
300        -1.858        -41.535
303        -2.092        -41.274
306        -2.364        -41.007
309        -2.678        -40.738
312        -3.039        -40.471
315        -3.452        -40.211
318        -3.924        -39.959
321        -4.462        -39.720
324        -5.075        -39.496
327        -5.772        -39.289
330        -6.562        -39.102
333        -7.457        -38.936
336        -8.464        -38.794
339        -9.582        -38.677
342       -10.785        -38.585
345       -11.990        -38.520
348       -13.009        -38.482
351       -13.545        -38.473
354       -13.365        -38.491
357       -12.529        -38.538
360       -11.253        -38.612

CPU RUN TIME FOR RUN   1 GEOMETRY   1 =      213.13 SECONDS

TOTAL CPU RUN TIME =      213.42 SECONDS
```

APPENDIX 2

OUTPUT FOR DESIGN EXAMPLE 2

```
     INPUT DATA

FREQ.(MHZ) =   300.000   WAVE(M) =     1.000   WIRE RADIUS(M) =  0.0010000
INTP= 10    INTD= 18   INT =    4   IFIL= 1

WIRE CONDUCTIVITY = -1.00 MEGAMHOS/M


GEOMETRY FOR THE   2 PLATES


                    PLATE NUMBER   1(RECTANGULAR)
NUMBER OF CORNERS =    4
MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
POLARIZATION INDICATOR =   3
GENERATING SIDE INDICATOR =   0

X,Y,Z COOR.(METERS) OF CORNER  1 =     0.00000     0.00000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  2 =     0.00000     0.00000     1.00000
X,Y,Z COOR.(METERS) OF CORNER  3 =     0.50000     0.00000     1.00000
X,Y,Z COOR.(METERS) OF CORNER  4 =     0.50000     0.00000     0.00000

COOR.(METERS) OF  10 NODES ON THIS PLATE


                    PLATE NUMBER   2(RECTANGULAR)
NUMBER OF CORNERS =    4
MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
POLARIZATION INDICATOR =   3
GENERATING SIDE INDICATOR =   0

X,Y,Z COOR.(METERS) OF CORNER  1 =     0.00000     0.00000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  2 =     0.00000     0.00000     1.00000
X,Y,Z COOR.(METERS) OF CORNER  3 =     0.00000     0.50000     1.00000
```

```
X,Y,Z COOR.(METERS) OF CORNER  4 =     0.00000     0.50000     0.00000

 COOR.(METERS) OF  10 MODES ON THIS PLATE


 COOR.(METERS) OF   4 OVERLAP MODES BETWEEN PLATE  1, SIDE 1 AND PLATE  2, SIDE 1


   LISTING OF LOADS AND GENERATORS


   NWR = NUMBER OF WIRE MODES =    0
   NPLTM = NUMBER OF PLATE MODES =   24
   NAT = NUMBER OF ATTACHMENT MODES =    0


 BACKSCATTERING, ISCAT =     1


                   ****  CROSS SECTION  (DB/WAVE**2)  ****     *******      PHASE  (DEG)       *******
THETA(DEG) PHI(DEG)  STTM       SPPM       STPM       SPTM       STTM       SPPM       STPM       SPTM
    90.0     0.0    -7.370     4.744    -56.149    -55.774    -133.51    -74.47    -88.55    -89.76
    90.0     3.0    -9.530     5.166    -55.888    -55.199    -110.89    -68.17    -84.27    -84.91
    90.0     6.0   -11.382     5.473    -55.698    -54.715     -75.23    -62.37    -80.60    -80.44
    90.0     9.0   -10.420     5.673    -55.576    -54.321     -31.29    -57.10    -77.50    -76.30
    90.0    12.0    -7.204     5.774    -55.530    -54.024      -0.26    -52.36    -75.00    -72.51
    90.0    15.0    -4.046     5.785    -55.562    -53.829      18.22    -48.19    -73.05    -69.06
    90.0    18.0    -1.445     5.716    -55.673    -53.736      30.42    -44.60    -71.66    -65.94
    90.0    21.0     0.660     5.581    -55.868    -53.752      39.33    -41.61    -70.80    -63.14
    90.0    24.0     2.370     5.394    -56.153    -53.882      46.23    -39.20    -70.48    -60.64
    90.0    27.0     3.759     5.172    -56.530    -54.129      51.72    -37.36    -70.68    -58.43
    90.0    30.0     4.876     4.935    -57.004    -54.498      56.11    -36.03    -71.40    -56.50
    90.0    33.0     5.757     4.705    -57.582    -54.995      59.56    -35.15    -72.62    -54.80
    90.0    36.0     6.424     4.500    -58.266    -55.622      62.17    -34.62    -74.34    -53.31
    90.0    39.0     6.891     4.340    -59.063    -56.386      64.01    -34.34    -76.56    -51.97
    90.0    42.0     7.170     4.238    -59.979    -57.289      65.09    -34.22    -79.28    -50.74
    90.0    45.0     7.266     4.203    -61.017    -58.338      65.45    -34.18    -82.52    -49.55
    90.0    48.0     7.180     4.239    -62.185    -59.537      65.08    -34.22    -86.28    -48.32
    90.0    51.0     6.912     4.342    -63.487    -60.890      63.98    -34.35    -90.60    -46.99
    90.0    54.0     6.456     4.503    -64.933    -62.407      62.13    -34.63    -95.52    -45.48
    90.0    57.0     5.802     4.708    -66.527    -64.106      59.50    -35.16   -101.16    -43.72
    90.0    60.0     4.938     4.940    -68.283    -66.009      56.03    -36.05   -107.64    -41.64
    90.0    63.0     3.840     5.177    -70.201    -68.179      51.63    -37.38   -115.22    -39.23
    90.0    66.0     2.478     5.399    -72.296    -70.736      46.14    -39.23   -124.29    -36.30
    90.0    69.0     0.805     5.587    -74.560    -73.938      39.25    -41.64   -135.46    -32.46
    90.0    72.0    -1.248     5.722    -76.930    -78.458      30.41    -44.63   -149.86    -25.83
    90.0    75.0    -3.770     5.791    -79.258    -86.394      18.42    -48.22   -168.60     -3.03
    90.0    78.0    -6.815     5.780    -81.123    -87.113       0.62    -52.40    167.31    102.15
    90.0    81.0    -9.977     5.680    -82.000    -79.366     -28.57    -57.14    139.96    125.98
    90.0    84.0   -11.258     5.480    -81.840    -75.262     -70.73    -62.42    113.96    130.73
    90.0    87.0    -9.716     5.173    -81.114    -72.589    -107.37    -68.22     91.70    131.67
    90.0    90.0    -7.624     4.752    -80.232    -70.672    -131.18    -74.52     72.84    131.12
```

90.0	93.0	-5.963	4.206	-79.359	-69.232	-148.14	-81.30	56.00	129.68
90.0	96.0	-4.769	3.526	-78.543	-68.129	-162.10	-88.58	40.41	127.72
90.0	99.0	-3.936	2.698	-77.764	-67.277	-174.76	-96.36	25.48	125.44
90.0	102.0	-3.369	1.703	-76.978	-66.620	173.09	-104.70	11.14	122.99
90.0	105.0	-2.993	0.517	-76.172	-66.124	161.08	-113.72	-2.61	120.46
90.0	108.0	-2.747	-0.895	-75.329	-65.759	148.99	-123.65	-15.60	117.89
90.0	111.0	-2.582	-2.582	-74.456	-65.497	136.74	-134.93	-27.67	115.36
90.0	114.0	-2.456	-4.598	-73.551	-65.324	124.31	-148.46	-38.73	112.88
90.0	117.0	-2.334	-6.965	-72.632	-65.229	111.74	-166.09	-48.73	110.47
90.0	120.0	-2.189	-9.441	-71.692	-65.201	99.11	168.64	-57.67	108.12
90.0	123.0	-2.000	-10.886	-70.742	-65.240	86.51	133.18	-65.64	105.84
90.0	126.0	-1.756	-10.006	-69.774	-65.341	74.04	96.88	-72.79	103.65
90.0	129.0	-1.456	-7.872	-68.792	-65.512	61.80	70.66	-79.22	101.63
90.0	132.0	-1.101	-5.748	-67.797	-65.745	49.85	52.79	-85.12	99.77
90.0	135.0	-0.701	-3.940	-66.798	-66.042	38.22	39.53	-90.65	98.16
90.0	138.0	-0.264	-2.431	-65.803	-66.395	26.91	28.78	-95.94	96.88
90.0	141.0	0.199	-1.160	-64.825	-66.795	15.91	19.52	-101.07	95.93
90.0	144.0	0.676	-0.072	-63.875	-67.225	5.19	11.18	-106.11	95.34
90.0	147.0	1.157	0.872	-62.970	-67.677	-5.30	3.45	-111.09	95.03
90.0	150.0	1.634	1.699	-62.116	-68.139	-15.57	-3.88	-115.99	94.83
90.0	153.0	2.096	2.425	-61.327	-68.613	-25.67	-10.95	-120.82	94.51
90.0	156.0	2.536	3.061	-60.606	-69.118	-35.60	-17.87	-125.54	93.71
90.0	159.0	2.947	3.610	-59.961	-69.698	-45.37	-24.71	-130.12	92.15
90.0	162.0	3.320	4.073	-59.396	-70.406	-54.99	-31.53	-134.53	89.36
90.0	165.0	3.649	4.447	-58.912	-71.330	-64.46	-38.38	-138.73	84.81
90.0	168.0	3.927	4.727	-58.512	-72.540	-73.76	-45.29	-142.69	77.60
90.0	171.0	4.146	4.905	-58.200	-74.084	-82.89	-52.30	-146.41	66.04
90.0	174.0	4.301	4.973	-57.983	-75.691	-91.82	-59.46	-149.86	47.02
90.0	177.0	4.385	4.923	-57.846	-76.561	-100.54	-66.83	-153.02	18.63
90.0	180.0	4.392	4.744	-57.808	-75.393	-109.05	-74.47	-155.90	-11.79
90.0	183.0	4.315	4.429	-57.869	-73.025	-117.34	-82.50	-158.48	-33.85
90.0	186.0	4.151	3.973	-58.032	-70.652	-125.40	-91.10	-160.80	-47.96
90.0	189.0	3.893	3.378	-58.300	-68.663	-133.24	-100.48	-162.86	-57.59
90.0	192.0	3.541	2.664	-58.683	-67.077	-140.87	-110.98	-164.71	-64.57
90.0	195.0	3.094	1.883	-59.189	-65.856	-148.31	-122.98	-166.38	-70.01
90.0	198.0	2.557	1.141	-59.824	-64.953	-155.59	-136.78	-167.94	-74.48
90.0	201.0	1.939	0.600	-60.599	-64.340	-162.73	-152.27	-169.49	-78.26
90.0	204.0	1.260	0.423	-61.528	-63.993	-169.75	-168.54	-171.17	-81.53
90.0	207.0	0.545	0.660	-62.614	-63.897	-176.63	176.03	-173.17	-84.46
90.0	210.0	-0.167	1.197	-63.865	-64.042	176.74	162.84	-175.78	-87.19
90.0	213.0	-0.830	1.849	-65.267	-64.417	170.52	152.50	-179.38	-89.81
90.0	216.0	-1.398	2.464	-66.782	-65.025	165.05	144.95	175.50	-92.55
90.0	219.0	-1.828	2.947	-68.300	-65.855	160.71	139.87	168.27	-95.58
90.0	222.0	-2.095	3.250	-69.619	-66.892	157.94	136.95	158.82	-99.22
90.0	225.0	-2.183	3.354	-70.507	-68.106	157.03	136.00	147.98	-103.78
90.0	228.0	-2.090	3.252	-70.861	-69.440	158.11	136.96	137.77	-109.61
90.0	231.0	-1.818	2.950	-70.815	-70.790	161.03	139.89	130.12	-117.02
90.0	234.0	-1.382	2.469	-70.616	-72.010	165.49	144.98	125.73	-125.84
90.0	237.0	-0.808	1.857	-70.452	-72.976	171.04	152.53	124.40	-135.29
90.0	240.0	-0.141	1.208	-70.428	-73.656	177.29	162.87	125.54	-144.09
90.0	243.0	0.574	0.675	-70.597	-74.159	-176.06	176.05	128.68	-151.16
90.0	246.0	1.288	0.440	-70.964	-74.638	-169.19	-168.53	133.47	-156.19
90.0	249.0	1.966	0.617	-71.542	-75.238	-162.18	-152.30	139.73	-159.33
90.0	252.0	2.579	1.157	-72.323	-76.051	-155.06	-136.83	147.35	-161.14
90.0	255.0	3.111	1.897	-73.299	-77.131	-147.79	-123.04	156.37	-162.34
90.0	258.0	3.552	2.676	-74.480	-78.497	-140.37	-111.04	166.98	-163.51
90.0	261.0	3.898	3.388	-75.824	-80.137	-132.74	-100.54	179.55	-165.60
90.0	264.0	4.149	3.982	-77.315	-82.004	-124.91	-91.15	-165.31	-169.49

```
 90.0  267.0     4.307      4.437    -78.816    -83.880   -116.85    -82.56   -146.79   -176.14
 90.0  270.0     4.378      4.752    -80.074    -85.497   -108.56    -74.52   -124.17    174.74
 90.0  273.0     4.366      4.930    -80.685    -86.613   -100.04    -66.87    -98.13    165.02
 90.0  276.0     4.277      4.980    -80.420    -87.368    -91.30    -59.51    -71.54    157.09
 90.0  279.0     4.118      4.912    -79.459    -88.149    -82.36    -52.34    -47.45    152.01
 90.0  282.0     3.895      4.734    -78.175    -89.314    -73.21    -45.32    -26.82    148.28
 90.0  285.0     3.616      4.454    -76.825    -91.093    -63.89    -38.41     -8.93    142.61
 90.0  288.0     3.286      4.080    -75.523    -93.294    -54.40    -31.56      7.16    128.29
 90.0  291.0     2.913      3.617    -74.294    -93.777    -44.75    -24.74     22.18    100.43
 90.0  294.0     2.504      3.067    -73.148    -91.021    -34.95    -17.90     36.54     79.14
 90.0  297.0     2.067      2.432    -72.070    -87.552    -25.00    -10.98     50.58     74.62
 90.0  300.0     1.609      1.706    -71.049    -84.586    -14.89     -3.91     64.45     79.32
 90.0  303.0     1.139      0.879    -70.070    -82.089     -4.60      3.42     78.26     88.51
 90.0  306.0     0.666     -0.064    -69.122    -79.901      5.90     11.15     92.02    100.18
 90.0  309.0     0.200     -1.152    -68.197    -77.896     16.62     19.48    105.71    113.10
 90.0  312.0    -0.249     -2.423    -67.284    -76.005     27.62     28.73    119.28    126.51
 90.0  315.0    -0.671     -3.932    -66.381    -74.172     38.90     39.46    132.66    139.93
 90.0  318.0    -1.055     -5.741    -65.485    -72.408     50.49     52.69    145.75    152.96
 90.0  321.0    -1.391     -7.868    -64.595    -70.699     62.38     70.51    158.50    165.37
 90.0  324.0    -1.672    -10.013    -63.717    -69.063     74.54     96.68    170.81    177.08
 90.0  327.0    -1.897    -10.913    -62.858    -67.504     86.91    133.01   -177.39   -171.93
 90.0  330.0    -2.068     -9.475    -62.022    -66.026     99.39    168.59   -166.15   -161.69
 90.0  333.0    -2.197     -6.993    -61.216    -64.632    111.89   -166.06   -155.51   -152.11
 90.0  336.0    -2.305     -4.620    -60.446    -63.324    124.31   -148.40   -145.49   -143.19
 90.0  339.0    -2.417     -2.599    -59.718    -62.097    136.59   -134.87   -136.10   -134.83
 90.0  342.0    -2.570     -0.910    -59.039    -60.954    148.68   -123.59   -127.38   -127.03
 90.0  345.0    -2.805      0.505    -58.410    -59.889    160.60   -113.66   -119.29   -119.73
 90.0  348.0    -3.169      1.693    -57.837    -58.906    172.43   -104.64   -111.87   -112.91
 90.0  351.0    -3.722      2.688    -57.322    -58.001   -175.65    -96.30   -105.09   -106.51
 90.0  354.0    -4.538      3.517    -56.874    -57.184   -163.28    -88.52    -98.97   -100.56
 90.0  357.0    -5.715      4.198    -56.476    -56.433   -149.76    -81.25    -93.44    -94.96
 90.0  360.0    -7.370      4.744    -56.149    -55.774   -133.51    -74.47    -88.55    -89.75

CPU RUN TIME FOR RUN   1 GEOMETRY   1 =      188.37 SECONDS


TOTAL CPU RUN TIME =      188.88 SECONDS
```

APPENDIX 3

OUTPUT FOR DESIGN EXAMPLE 3

```
      INPUT DATA

 FREQ.(MHZ) =   300.000   WAVE(M) =     1.000   WIRE RADIUS(M) =  0.0010000
 INTP=  10    INTD=  18   INT =     4   IFIL= 1


 WIRE CONDUCTIVITY = -1.00 MEGAMHOS/M


 GEOMETRY FOR THE    2 PLATES


                     PLATE NUMBER    1(RECTANGULAR)
 NUMBER OF CORNERS =    4
 MAXIMUM SEGMENT SIZE   (METERS) =     0.25000
 POLARIZATION INDICATOR =    3
 GENERATING SIDE INDICATOR =    0

 X,Y,Z COOR.(METERS) OF CORNER  1 =      0.00000      0.00000      0.00000
 X,Y,Z COOR.(METERS) OF CORNER  2 =      0.00000      0.00000      1.00000
 X,Y,Z COOR.(METERS) OF CORNER  3 =      0.50000      0.00000      1.00000
 X,Y,Z COOR.(METERS) OF CORNER  4 =      0.50000      0.00000      0.00000

 COOR.(METERS) OF  10 MODES ON THIS PLATE



                     PLATE NUMBER    2(RECTANGULAR)
 NUMBER OF CORNERS =    4
 MAXIMUM SEGMENT SIZE   (METERS) =     0.25000
 POLARIZATION INDICATOR =    3
 GENERATING SIDE INDICATOR =    0

 X,Y,Z COOR.(METERS) OF CORNER  1 =      0.00000      0.00000      0.00000
 X,Y,Z COOR.(METERS) OF CORNER  2 =      0.00000      0.00000      1.00000
 X,Y,Z COOR.(METERS) OF CORNER  3 =      0.00000      0.50000      1.00000
```

```
X,Y,Z COOR.(METERS) OF CORNER  4 =      0.00000     0.50000     0.00000

 COOR.(METERS) OF  10 MODES ON THIS PLATE


 COOR.(METERS) OF   4 OVERLAP MODES BETWEEN PLATE  1, SIDE 1 AND PLATE  2, SIDE 1


   LISTING OF LOADS AND GENERATORS


   NWR = NUMBER OF WIRE MODES =    0
   NPLTM = NUMBER OF PLATE MODES =   24
   NAT = NUMBER OF ATTACHMENT MODES =    0



 BISTATIC SCATTERING, ISCAT =     2
 THETA INC.(DEG.) =   90.0
 PHI INC.(DEG.) =   45.0



                     ****  CROSS SECTION (DB/WAVE**2)  ****       *******    PHASE  (DEG)      *******
THETA(DEG) PHI(DEG)  STTM       SPPM      STPM      SPTM          STTM      SPPM      STPM       SPTM
     90.0     0.0    1.267      1.252    -58.866   -58.961        27.28    -70.30   -138.83     -47.99
     90.0     3.0    2.032      1.588    -58.720   -58.568        32.00    -65.68   -133.42     -47.19
     90.0     6.0    2.745      1.914    -58.601   -58.221        36.46    -61.37   -128.17     -46.61
     90.0     9.0    3.407      2.227    -58.[illegible]0   -57.915        40.63    -57.38   -123.10     -46.22
     90.0    12.0    4.018      2.524    -58.450   -57.660        44.51    -53.70   -118.22     -46.02
     90.0    15.0    4.577      2.801    -58.424   -57.454        48.08    -50.33   -113.57     -45.98
     90.0    18.0    5.083      3.057    -58.436   -57.298        51.34    -47.28   -109.16     -46.08
     90.0    21.0    5.538      3.291    -58.489   -57.196        54.27    -44.54   -105.01     -46.31
     90.0    24.0    5.940      3.500    -58.589   -57.149        56.87    -42.12   -101.13     -46.64
     90.0    27.0    6.289      3.684    -58.738   -57.156        59.13    -40.02    -97.54     -47.06
     90.0    30.0    6.585      3.841    -58.942   -57.219        61.06    -38.24    -94.24     -47.54
     90.0    33.0    6.828      3.970    -59.208   -57.339        62.63    -36.78    -91.25     -48.05
     90.0    36.0    7.018      4.071    -59.540   -57.514        63.86    -35.65    -88.57     -48.55
     90.0    39.0    7.154      4.144    -59.946   -57.741        64.74    -34.83    -86.22     -49.00
     90.0    42.0    7.237      4.188    -60.435   -58.018        65.27    -34.35    -84.20     -49.35
     90.0    45.0    7.266      4.203    -61.017   -58.338        65.45    -34.18    -82.52     -49.55
     90.0    48.0    7.242      4.189    -61.703   -58.693        65.27    -34.35    -81.19     -49.53
     90.0    51.0    7.164      4.145    -62.311   -59.072        64.74    -34.84    -80.23     -49.26
     90.0    54.0    7.032      4.073    -63.461   -59.460        63.86    -35 65    -79.67     -48.67
     90.0    57.0    6.848      3.972    -64.583   -59.838        62.62    -36.79    -79.56     -47.76
     90.0    60.0    6.610      3.843    -65.920   -60.184        61.04    -38.25    -80.01     -46.52
     90.0    63.0    6.320      3.687    -67.536   -60.476        59.12    -40.03    -81.17     -45.03
     90.0    66.0    5.977      3.504    -69.537   -60.694        56.85    -42.13    -83.42     -43.39
     90.0    69.0    5.582      3.295    -72.108   -60.827        54.25    -44.55    -87.60     -41.77
     90.0    72.0    5.136      3.062    -75.593   -60.874        51.32    -47.29    -96.20     -40.34
     90.0    75.0    4.638      2.806    -80.413   -60.841        48.07    -50.34   -118.61     -39.28
     90.0    78.0    4.089      2.529    -82.856   -60.748        44.50    -53.71   -176.74     -38.70
```

90.0	81.0	3.490	2.233	-78.369	-60.615	40.63	-57.39	142.61	-38.70
90.0	84.0	2.840	1.921	-74.486	-60.465	36.46	-61.38	128.38	-39.28
90.0	87.0	2.141	1.595	-71.773	-60.318	32.01	-65.69	121.66	-40.43
90.0	90.0	1.392	1.259	-69.773	-60.193	27.31	-70.31	117.58	-42.08
90.0	93.0	0.594	0.917	-68.227	-60.103	22.36	-75.24	114.68	-44.18
90.0	96.0	-0.253	0.574	-66.987	-60.058	17.19	-80.49	112.39	-46.65
90.0	99.0	-1.150	0.234	-65.968	-60.066	11.85	-86.03	110.47	-49.40
90.0	102.0	-2.096	-0.095	-65.111	-60.132	6.38	-91.87	108.77	-52.37
90.0	105.0	-3.092	-0.410	-64.379	-60.261	0.84	-98.00	107.21	-55.48
90.0	108.0	-4.139	-0.702	-63.741	-60.456	-4.69	-104.39	105.75	-58.66
90.0	111.0	-5.236	-0.965	-63.177	-60.720	-10.11	-111.02	104.35	-61.84
90.0	114.0	-6.384	-1.194	-62.671	-61.054	-15.25	-117.86	102.98	-64.96
90.0	117.0	-7.579	-1.383	-62.208	-61.459	-19.91	-124.87	101.59	-67.94
90.0	120.0	-8.811	-1.529	-61.781	-61.938	-23.81	-132.00	100.17	-70.73
90.0	123.0	-10.055	-1.627	-61.381	-62.488	-26.58	-139.21	98.69	-73.25
90.0	126.0	-11.259	-1.679	-61.004	-63.109	-27.77	-146.43	97.12	-75.42
90.0	129.0	-12.322	-1.683	-60.648	-63.799	-26.98	-153.62	95.43	-77.18
90.0	132.0	-13.084	-1.644	-60.309	-64.552	-24.10	-160.74	93.60	-78.43
90.0	135.0	-13.369	-1.564	-59.990	-65.360	-19.79	-167.75	91.62	-79.13
90.0	138.0	-13.094	-1.448	-59.690	-66.211	-15.46	-174.60	89.47	-79.19
90.0	141.0	-12.338	-1.303	-59.411	-67.090	-12.53	178.71	87.15	-78.61
90.0	144.0	-11.277	-1.133	-59.155	-67.977	-11.67	172.21	84.64	-77.39
90.0	147.0	-10.072	-0.943	-58.925	-68.856	-12.81	165.91	81.95	-75.64
90.0	150.0	-8.824	-0.740	-58.724	-69.713	-15.53	159.81	79.08	-73.53
90.0	153.0	-7.589	-0.528	-58.555	-70.544	-19.39	153.91	76.03	-71.36
90.0	156.0	-6.391	-0.310	-58.419	-71.370	-24.01	148.22	72.80	-69.44
90.0	159.0	-5.241	-0.092	-58.321	-72.230	-29.12	142.72	69.41	-68.19
90.0	162.0	-4.141	0.124	-58.262	-73.177	-34.51	137.42	65.85	-68.03
90.0	165.0	-3.092	0.335	-58.245	-74.293	-40.02	132.32	62.13	-69.51
90.0	168.0	-2.095	0.539	-58.273	-75.652	-45.54	127.40	58.26	-73.59
90.0	171.0	-1.148	0.734	-58.347	-77.291	-51.00	122.68	54.23	-81.93
90.0	174.0	-0.251	0.918	-58.470	-78.948	-56.32	118.16	50.05	-97.73
90.0	177.0	0.597	1.091	-58.642	-79.808	-61.47	113.83	45.70	-122.90
90.0	180.0	1.395	1.252	-58.866	-78.578	-66.41	109.70	41.17	-150.01
90.0	183.0	2.143	1.399	-59.142	-76.094	-71.11	105.77	36.44	-168.83
90.0	186.0	2.842	1.534	-59.470	-73.545	-75.55	102.06	31.50	-179.68
90.0	189.0	3.490	1.656	-59.847	-71.319	-79.71	98.56	26.29	173.93
90.0	192.0	4.089	1.764	-60.270	-69.429	-83.57	95.29	20.80	170.17
90.0	195.0	4.636	1.861	-60.733	-67.834	-87.14	92.26	14.96	167.94
90.0	198.0	5.133	1.945	-61.226	-66.485	-90.38	89.48	8.73	166.68
90.0	201.0	5.578	2.018	-61.734	-65.349	-93.31	86.95	2.08	166.14
90.0	204.0	5.971	2.080	-62.239	-64.396	-95.90	84.69	-5.02	166.16
90.0	207.0	6.312	2.132	-62.715	-63.605	-98.16	82.70	-12.55	166.60
90.0	210.0	6.600	2.175	-63.136	-62.962	-100.08	81.00	-20.44	167.44
90.0	213.0	6.836	2.209	-63.479	-62.454	-101.66	79.60	-28.53	168.63
90.0	216.0	7.018	2.235	-63.726	-62.074	-102.88	78.50	-36.60	170.12
90.0	219.0	7.148	2.253	-63.873	-61.815	-103.76	77.71	-44.41	171.89
90.0	222.0	7.223	2.264	-63.930	-61.673	-104.28	77.23	-51.71	173.92
90.0	225.0	7.246	2.268	-63.919	-61.645	-104.45	77.08	-58.30	176.19
90.0	228.0	7.214	2.265	-63.869	-61.730	-104.27	77.23	-64.06	178.67
90.0	231.0	7.129	2.255	-63.808	-61.927	-103.73	77.71	-68.92	-178.67
90.0	234.0	6.991	2.237	-63.762	-62.239	-102.83	78.50	-72.89	-175.85
90.0	237.0	6.799	2.212	-63.753	-62.667	-101.59	79.60	-76.00	-172.92
90.0	240.0	6.554	2.178	-63.799	-63.216	-100.00	81.00	-78.31	-169.93
90.0	243.0	6.256	2.136	-63.912	-63.891	-98.05	82.70	-79.86	-166.95
90.0	246.0	5.904	2.085	-64.103	-64.699	-95.77	84.68	-80.70	-164.09
90.0	249.0	5.501	2.023	-64.382	-65.648	-93.14	86.94	-80.88	-161.45
90.0	252.0	5.045	1.950	-64.758	-66.749	-90.18	89.47	-80.41	-159.24

```
90.0  255.0     4.537     1.867   -65.240   -68.008    -86.89     92.26    -79.31   -157.72
90.0  258.0     3.977     1.771   -65.840   -69.424    -83.29     95.29    -77.57   -157.30
90.0  261.0     3.366     1.662   -66.572   -70.970    -79.36     98.56    -75.14   -158.56
90.0  264.0     2.705     1.541   -67.454   -72.566    -75.14    102.05    -71.93   -162.25
90.0  267.0     1.992     1.406   -68.511   -74.015    -70.62    105.76    -67.79   -169.02
90.0  270.0     1.229     1.259   -69.773   -75.002    -65.83    109.69    -62.42   -178.52
90.0  273.0     0.415     1.098   -71.280   -75.276    -60.79    113.82    -55.30    171.44
90.0  276.0    -0.450     0.926   -73.057   -74.917    -55.51    118.15    -45.43    163.69
90.0  279.0    -1.366     0.741   -75.057   -74.239    -50.04    122.67    -31.01    159.59
90.0  282.0    -2.333     0.546   -76.926   -73.507    -44.40    127.39     -9.31    158.94
90.0  285.0    -3.353     0.342   -77.676   -72.852    -38.67    132.30     19.80    161.01
90.0  288.0    -4.426     0.130   -76.642   -72.315    -32.92    137.40     48.20    165.17
90.0  291.0    -5.555    -0.086   -74.693   -71.895    -27.26    142.70     69.06    170.97
90.0  294.0    -6.741    -0.305   -72.705   -71.565    -21.83    148.19     83.52    178.13
90.0  297.0    -7.983    -0.523   -70.949   -71.291    -16.85    153.88     94.23   -173.56
90.0  300.0    -9.276    -0.736   -69.446   -71.032    -12.61    159.78    102.82   -164.29
90.0  303.0   -10.601    -0.940   -68.158   -70.744     -9.52    165.88    110.20   -154.30
90.0  306.0   -11.910    -1.130   -67.047   -70.390     -8.11    172.18    116.88   -143.83
90.0  309.0   -13.103    -1.301   -66.078   -69.942     -8.95    178.67    123.12   -133.20
90.0  312.0   -13.995    -1.448   -65.226   -69.393    -12.30   -174.64    129.11   -122.77
90.0  315.0   -14.360    -1.564   -64.471   -68.751    -17.56   -167.78    134.94   -112.84
90.0  318.0   -14.071    -1.645   -63.797   -68.037    -23.00   -160.78    140.69   -103.63
90.0  321.0   -13.222    -1.686   -63.191   -67.275    -26.78   -153.66    146.40    -95.26
90.0  324.0   -12.038    -1.682   -62.644   -66.492    -28.10   -146.46    152.11    -87.79
90.0  327.0   -10.715    -1.632   -62.148   -65.709    -27.12   -139.24    157.84    -81.17
90.0  330.0    -9.367    -1.534   -61.697   -64.938    -24.39   -132.03    163.58    -75.38
90.0  333.0    -8.050    -1.390   -61.285   -64.190    -20.44   -124.89    169.36    -70.33
90.0  336.0    -6.785    -1.201   -60.909   -63.471    -15.69   -117.88    175.16    -65.96
90.0  339.0    -5.579    -0.973   -60.563   -62.783    -10.46   -111.04   -179.01    -62.19
90.0  342.0    -4.434    -0.710   -60.247   -62.130     -4.97   -104.40   -173.17    -58.96
90.0  345.0    -3.346    -0.418   -59.957   -61.510      0.63    -98.00   -167.34    -56.21
90.0  348.0    -2.316    -0.104   -59.692   -60.926      6.23    -91.87   -161.52    -53.88
90.0  351.0    -1.341     0.226   -59.450   -60.378     11.75    -86.03   -155.74    -51.94
90.0  354.0    -0.419     0.566   -59.232   -59.875     17.12    -80.48   -150.01    -50.36
90.0  357.0     0.450     0.910   -59.038   -59.394     22.31    -75.24   -144.37    -49.02
90.0  360.0     1.267     1.252   -58.866   -58.961     27.28    -70.30   -138.83    -47.99

CPU RUN TIME FOR RUN   1 GEOMETRY   1 =      186.45 SECONDS


TOTAL CPU RUN TIME =      186.96 SECONDS
```

APPENDIX 4

OUTPUT FOR DESIGN EXAMPLE 4

```
      INPUT DATA

   FREQ.(MHZ) =   300.000   WAVE(M) =      1.000   WIRE RADIUS(M) =  0.0010000
   INTP=   8   INTD=  18   INT =    4   IFIL= 1


   WIRE CONDUCTIVITY = -1.00 MEGAMHOS/M


   GEOMETRY FOR THE   3 PLATES


                     PLATE NUMBER   1(RECTANGULAR)
   NUMBER OF CORNERS =   4
   MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
   POLARIZATION INDICATOR =   3
   GENERATING SIDE INDICATOR =   0

  X,Y,Z COOR.(METERS) OF CORNER  1 =     0.50000    -0.50000    -1.00000
  X,Y,Z COOR.(METERS) OF CORNER  2 =     0.50000    -0.50000     0.00000
  X,Y,Z COOR.(METERS) OF CORNER  3 =    -0.50000    -0.50000     0.00000
  X,Y,Z COOR.(METERS) OF CORNER  4 =    -0.50000    -0.50000    -1.00000

  COOR.(METERS) OF  24 NODES ON THIS PLATE


                     PLATE NUMBER   2(RECTANGULAR)
   NUMBER OF CORNERS =   4
   MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
   POLARIZATION INDICATOR =   3
   GENERATING SIDE INDICATOR =   0

  X,Y,Z COOR.(METERS) OF CORNER  1 =     0.50000    -0.50000     0.00000
  X,Y,Z COOR.(METERS) OF CORNER  2 =     0.50000     0.50000     0.00000
  X,Y,Z COOR.(METERS) OF CORNER  3 =    -0.50000     0.50000     0.00000
```

```
   2     2      3    0.12500E+00



 GEOMETRY FOR THE  1 ATTACHMENT POINTS

 I  SEGMENT  END  PLATE      B(M)

 1      1     0      2     0.20000


   LISTING OF LOADS AND GENERATORS

    0.1000E+01   0.0000E+00 VOLTS AT ATTACHMENT  1


   NWR = NUMBER OF WIRE MODES =    1
   NPLTM = NUMBER OF PLATE MODES =   64
   NAT = NUMBER OF ATTACHMENT MODES =    1



 INPUT ADMITTANCE(MHOS) =   0.019068 J  -0.000726
 INPUT IMPEDANCE(OHMS) =     52.368 J      1.994
 EFFICIENCY(PERCENT) = 100.000



 CPU RUN TIME FOR RUN   1 GEOMETRY   1 =      342.63 SECONDS



       INPUT DATA

 FREQ.(MHZ) =   300.000   WAVE(M) =     1.000   WIRE RADIUS(M) =  0.0010000
 INTP=   4   INTD=  18   INT =    4   IFIL= 1


 WIRE CONDUCTIVITY = -1.00 MEGAMHOS/M



 GEOMETRY FOR THE   3 PLATES



                      PLATE NUMBER   1(RECTANGULAR)
 NUMBER OF CORNERS =   4
 MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
 POLARIZATION INDICATOR =   3
 GENERATING SIDE INDICATOR =   0

 X,Y,Z COOR.(METERS) OF CORNER  1 =     0.50000    -0.50000    -1.00000
 X,Y,Z COOR.(METERS) OF CORNER  2 =     0.50000    -0.50000     0.00000
 X,Y,Z COOR.(METERS) OF CORNER  3 =    -0.50000    -0.50000     0.00000
 X,Y,Z COOR.(METERS) OF CORNER  4 =    -0.50000    -0.50000    -1.00000
```

```
X,Y,Z COOR.(METERS) OF CORNER  4 =    -0.50000    -0.50000     0.00000

 COOR.(METERS) OF  24 NODES ON THIS PLATE



                        PLATE NUMBER   3(RECTANGULAR)
 NUMBER OF CORNERS =   4
 MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
 POLARIZATION INDICATOR =   3
 GENERATING SIDE INDICATOR =   0

X,Y,Z COOR.(METERS) OF CORNER  1 =     0.25000     0.50000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  2 =    -0.25000     0.50000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  3 =    -0.25000     0.50000     1.00000
X,Y,Z COOR.(METERS) OF CORNER  4 =     0.25000     0.50000     1.00000

 COOR.(METERS) OF  10 NODES ON THIS PLATE


 COOR.(METERS) OF   4 OVERLAP NODES BETWEEN PLATE  1, SIDE 2 AND PLATE  2, SIDE 4


 COOR.(METERS) OF   2 OVERLAP NODES BETWEEN PLATE  2, SIDE 2 AND PLATE  3, SIDE 1




    3 POINTS ON THE WIRE
  I        X (I)           Y (I)          Z (I)

  1   0.0000E+00     0.0000E+00     0.0000E+00
  2   0.0000E+00     0.0000E+00     0.1250E+00
  3   0.0000E+00     0.0000E+00     0.2500E+00


  NODES ON THE WIRE STRUCTURE

  MAXIMUM NUMBER OF NODES AT ONE POINT =  1
  MINIMUM NUMBER OF NODES AT ONE POINT =  1
  NUMBER OF WIRE NODES =   1


    2 SEGMENTS ON THE WIRE
  J     IA(J)   IB(J)    D(J)(M)

  1      1        2     0.12500E+00
```

```
COOR.(METERS) OF  24 NODES ON THIS PLATE


                         PLATE NUMBER   2(RECTANGULAR)
NUMBER OF CORNERS =   4
MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
POLARIZATION INDICATOR =   3
GENERATING SIDE INDICATOR =   0

X,Y,Z COOR.(METERS) OF CORNER  1 =    0.50000   -0.50000    0.00000
X,Y,Z COOR.(METERS) OF CORNER  2 =    0.50000    0.50000    0.00000
X,Y,Z COOR.(METERS) OF CORNER  3 =   -0.50000    0.50000    0.00000
X,Y,Z COOR.(METERS) OF CORNER  4 =   -0.50000   -0.50000    0.00000

COOR.(METERS) OF  24 NODES ON THIS PLATE


                         PLATE NUMBER   3(RECTANGULAR)
NUMBER OF CORNERS =   4
MAXIMUM SEGMENT SIZE  (METERS) =    0.25000
POLARIZATION INDICATOR =   3
GENERATING SIDE INDICATOR =   0

X,Y,Z COOR.(METERS) OF CORNER  1 =    0.25000    0.50000    0.00000
X,Y,Z COOR.(METERS) OF CORNER  2 =   -0.25000    0.50000    0.00000
X,Y,Z COOR.(METERS) OF CORNER  3 =   -0.25000    0.50000    1.00000
X,Y,Z COOR.(METERS) OF CORNER  4 =    0.25000    0.50000    1.00000

COOR.(METERS) OF  10 NODES ON THIS PLATE


COOR.(METERS) OF   4 OVERLAP NODES BETWEEN PLATE  1, SIDE 2 AND PLATE  2, SIDE 4


COOR.(METERS) OF   2 OVERLAP NODES BETWEEN PLATE  2, SIDE 2 AND PLATE  3, SIDE 1



   3 POINTS ON THE WIRE
 I       X (I)          Y (I)          Z (I)

 1   0.0000E+00     0.3000E+00     0.0000E+00
 2   0.0000E+00     0.3000E+00     0.1250E+00
 3   0.0000E+00     0.3000E+00     0.2500E+00
```

```
  MODES ON THE WIRE STRUCTURE

  MAXIMUM NUMBER OF MODES AT ONE POINT =  1
  MINIMUM NUMBER OF MODES AT ONE POINT =  1
  NUMBER OF WIRE MODES =   1



    2 SEGMENTS ON THE WIRE
  J     IA(J)   IB(J)    D(J)(M)

  1      1       2     0.12500E+00
  2      2       3     0.12500E+00



GEOMETRY FOR THE  1 ATTACHMENT POINTS

I  SEGMENT  END  PLATE      B(M)

1     1      0      2      0.20000


  LISTING OF LOADS AND GENERATORS

   0.1000E+01   0.0000E+00 VOLTS AT ATTACHMENT  1


  NWR = NUMBER OF WIRE MODES =    1
  NPLTM = NUMBER OF PLATE MODES =   64
  NAT = NUMBER OF ATTACHMENT MODES =    1


INPUT ADMITTANCE(MHOS) =   0.015535 J  -0.012869
INPUT IMPEDANCE(OHMS) =     38.174 J     31.623
EFFICIENCY(PERCENT) = 100.000


CPU RUN TIME FOR RUN  1 GEOMETRY   2 =       89.03 SECONDS


TOTAL CPU RUN TIME =      432.05 SECONDS
```

APPENDIX 5

OUTPUT FOR DESIGN EXAMPLE 5

```
      INPUT DATA

FREQ.(MHZ) =   300.000   WAVE(M) =     1.000   WIRE RADIUS(M) =  0.0010000
INTP=  10   INTD=  18   INT =    4   IFIL= 1


WIRE CONDUCTIVITY = -1.00 MEGAMHOS/M


GEOMETRY FOR THE   1 PLATES


                        PLATE NUMBER   1(POLYGONAL)
NUMBER OF CORNERS =   8
MAXIMUM SEGMENT SIZE  (METERS) =     0.25000
POLARIZATION INDICATOR =   3
GENERATING SIDE INDICATOR =   0

X,Y,Z COOR.(METERS) OF CORNER  1 =     0.00000     0.30000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  2 =     0.21200     0.21200     0.00000
X,Y,Z COOR.(METERS) OF CORNER  3 =     0.30000     0.00000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  4 =     0.21200    -0.21200     0.00000
X,Y,Z COOR.(METERS) OF CORNER  5 =     0.00000    -0.30000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  6 =    -0.21200    -0.21200     0.00000
X,Y,Z COOR.(METERS) OF CORNER  7 =    -0.30000     0.00000     0.00000
X,Y,Z COOR.(METERS) OF CORNER  8 =    -0.21200     0.21200     0.00000

COOR.(METERS) OF  12 MODES ON THIS PLATE

MONOPOLE  MODE    X1       Y1       Z1       X2       Y2       Z2       X3       Y3       Z3       X4       Y4       Z4

   A       1   0.24133  0.14133  0.00000  0.21200  0.21200  0.00000  0.00000  0.30000  0.00000  0.07067  0.12933  0.00000
   B       1   0.24133  0.14133  0.00000  0.27067  0.07067  0.00000  0.14133 -0.04133  0.00000  0.07067  0.12933  0.00000

   A       2   0.27067  0.07067  0.00000  0.24133  0.14133  0.00000  0.07067  0.12933  0.00000  0.14133 -0.04133  0.00000
   B       2   0.27067  0.07067  0.00000  0.30000  0.00000  0.00000  0.21200 -0.21200  0.00000  0.14133 -0.04133  0.00000
```

```
A      3    0.07067  0.12933  0.00000  0.00000  0.30000  0.00000 -0.21200  0.21200  0.00000 -0.14133  0.04133  0.00000
B      3    0.07067  0.12933  0.00000  0.14133 -0.04133  0.00000 -0.07067 -0.12933  0.00000 -0.14133  0.04133  0.00000

A      4    0.14133 -0.04133  0.00000  0.07067  0.12933  0.00000 -0.14133  0.04133  0.00000 -0.07067 -0.12933  0.00000
B      4    0.14133 -0.04133  0.00000  0.21200 -0.21200  0.00000  0.00000 -0.30000  0.00000 -0.07067 -0.12933  0.00000

A      5   -0.14133  0.04133  0.00000 -0.21200  0.21200  0.00000 -0.30000  0.00000  0.00000 -0.27067 -0.07067  0.00000
B      5   -0.14133  0.04133  0.00000 -0.07067 -0.12933  0.00000 -0.24133 -0.14133  0.00000 -0.27067 -0.07067  0.00000

A      6   -0.07067 -0.12933  0.00000 -0.14133  0.04133  0.00000 -0.27067 -0.07067  0.00000 -0.24133 -0.14133  0.00000
B      6   -0.07067 -0.12933  0.00000  0.00000 -0.30000  0.00000 -0.21200 -0.21200  0.00000 -0.24133 -0.14133  0.00000

A      7    0.00000  0.30000  0.00000  0.21200  0.21200  0.00000  0.24133  0.14133  0.00000  0.07067  0.12933  0.00000
B      7    0.00000  0.30000  0.00000 -0.21200  0.21200  0.00000 -0.14133  0.04133  0.00000  0.07067  0.12933  0.00000

A      8   -0.21200  0.21200  0.00000  0.00000  0.30000  0.00000  0.07067  0.12933  0.00000 -0.14133  0.04133  0.00000
B      8   -0.21200  0.21200  0.00000 -0.30000  0.00000  0.00000 -0.27067 -0.07067  0.00000 -0.14133  0.04133  0.00000

A      9    0.07067  0.12933  0.00000  0.24133  0.14133  0.00000  0.27067  0.07067  0.00000  0.14133 -0.04133  0.00000
B      9    0.07067  0.12933  0.00000 -0.14133  0.04133  0.00000 -0.07067 -0.12933  0.00000  0.14133 -0.04133  0.00000

A     10   -0.14133  0.04133  0.00000  0.07067  0.12933  0.00000  0.14133 -0.04133  0.00000 -0.07067 -0.12933  0.00000
B     10   -0.14133  0.04133  0.00000 -0.27067 -0.07067  0.00000 -0.24133 -0.14133  0.00000 -0.07067 -0.12933  0.00000

A     11    0.14133 -0.04133  0.00000  0.27067  0.07067  0.00000  0.30000  0.00000  0.00000  0.21200 -0.21200  0.00000
B     11    0.14133 -0.04133  0.00000 -0.07067 -0.12933  0.00000  0.00000 -0.30000  0.00000  0.21200 -0.21200  0.00000

A     12   -0.07067 -0.12933  0.00000  0.14133 -0.04133  0.00000  0.21200 -0.21200  0.00000  0.00000 -0.30000  0.00000
B     12   -0.07067 -0.12933  0.00000 -0.24133 -0.14133  0.00000 -0.21200 -0.21200  0.00000  0.00000 -0.30000  0.00000


LISTING OF LOADS AND GENERATORS


NWR = NUMBER OF WIRE NODES =    0
NPLTN = NUMBER OF PLATE NODES =   12
NAT = NUMBER OF ATTACHMENT NODES =    0


TOTAL CPU RUN TIME =       3.63 SECONDS
```

APPENDIX 6

SUBROUTINE PLPLCK

```
      SUBROUTINE PLPLCK(PCN,IPL,ICN,N,TOUCH,NP,IOK)
      DIMENSION PCN(3,ICN,IPL)
      IF (N.EQ.3)RETURN
      DS1=SQRT((PCN(1,2,NP)-PCN(1,1,NP))**2+(PCN(2,2,NP)
     &-PCN(2,1,NP))**2+(PCN(3,2,NP)-PCN(3,1,NP))**2)
      DSN=SQRT((PCN(1,N,NP)-PCN(1,1,NP))**2+(PCN(2,N,NP)
     &-PCN(2,1,NP))**2+(PCN(3,N,NP)-PCN(3,1,NP))**2)
      CAX=(PCN(1,2,NP)-PCN(1,1,NP))/DS1
      CBX=(PCN(2,2,NP)-PCN(2,1,NP))/DS1
      CGX=(PCN(3,2,NP)-PCN(3,1,NP))/DS1
      CAT=(PCN(1,N,NP)-PCN(1,1,NP))/DSN
      CBT=(PCN(2,N,NP)-PCN(2,1,NP))/DSN
      CGT=(PCN(3,N,NP)-PCN(3,1,NP))/DSN
      CC=CAX*CAT+CBX*CBT+CGX*CGT
      IF(ABS(CC).LT.0.99999)GOTO 15
      IOK=0
      WRITE(6,100)N,NP
100   FORMAT(' ***** WARNING *****  CORNERS 1 AND',I3,' OF PLATE',I3
     &,' ARE PARALLEL')
      RETURN
15    SS=SQRT(1.-CC*CC)
      CAZ=(CBX*CGT-CBT*CGX)/SS
      CBZ=(CGX*CAT-CGT*CAX)/SS
      CGZ=(CAX*CBT-CAT*CBX)/SS
      CAY=CBZ*CGX-CBX*CGZ
      CBY=CGZ*CAX-CGX*CAZ
      CGY=CAZ*CBX-CAX*CBZ
      DO 20 IC=3,N-1
      XP=CAX*(PCN(1,IC,NP)-PCN(1,1,NP))+CBX*(PCN(2,IC,NP)-PCN(2,1,NP))
     &+CGX*(PCN(3,IC,NP)-PCN(3,1,NP))
      YP=CAY*(PCN(1,IC,NP)-PCN(1,1,NP))+CBY*(PCN(2,IC,NP)-PCN(2,1,NP))
     &+CGY*(PCN(3,IC,NP)-PCN(3,1,NP))
      ZP=CAZ*(PCN(1,IC,NP)-PCN(1,1,NP))+CBZ*(PCN(2,IC,NP)-PCN(2,1,NP))
     &+CGZ*(PCN(3,IC,NP)-PCN(3,1,NP))
      IF(ZP.EQ.0.)GOTO 20
      IF(ZP.GT.TOUCH/2.)GOTO 25
      PCN(1,IC,NP)=CAX*XP+CAY*YP+PCN(1,1,NP)
      PCN(2,IC,NP)=CBX*XP+CBY*YP+PCN(2,1,NP)
      PCN(2,IC,NP)=CGX*XP+CGY*YP+PCN(3,1,NP)
      GOTO 20
25    IOK=0
      WRITE(6,105)NP,IC,ZP,N
105   FORMAT(' ***** WARNING *****  PLATE',I3,', CORNER',I3,' IS'
     &,E10.3,' METERS OUT FOR THE PLANE FORMED BY CORNERS 1, 2, AND',I3)
20    CONTINUE
      END
```

APPENDIX 7

SUBROUTINE PLATE3

```
        SUBROUTINE PLATE3(PC,NC,ICN,NP,NDNPLT,PA,PB,IPLM,SEGM
     &,IQUAD,WV,IRE,IP,MPL1,MPL2,IOK,NM12,NM23,IGS)
C THIS SUBROUTINE BREAKS UP A POLYGONAL PLATE INTO QUADRALATERAL MODES.
C PC = COORDINATES OF PLATE
C NC = NUMBER OF CORNERS IN PLATE
C NP = PLATE NUMBER
C WV = WAVELENGTH IN METERS
C MPL1 = NUMBER OF MODES WITH FIRST POLARIZATION
C MPL2 = TOTAL NUMBER OF MODES ON PLATE
C IOK = STATUS INDICATOR; = 0 IF RUN IS TO ABORT
C IGS = GENERATING SIDE NUMBER, = 0 IF GENERATING SIDE TO BE CHOSEN BY PLATE3
C MDME = MAXIMUM MODE DIFFERENCE FOR POLYGON WITH EVEN NUMBER OF SIDES
C MDMO = MAXIMUM MODE DIFFERENCE FOR POLYGON WITH ODD NUMBER OF SIDES
C EVSD = .TRUE. IF POLYGON HAS EVEN NUMBER OF SIDES
        DIMENSION PC(3,ICN),NDNPLT(1),PA(IPLM,4,3),PB(IPLM,4,3)
        DIMENSION GPT(50,2,3),CA(3),CB(3),CG(3),CD(3),CE(3)
        LOGICAL EVSD
C SET MAXIMUM MODE DIFFERENCE PARAMETERS
        MDME=5
        MDMO=2
        MPL1=0
        MPL2=0
        NM12=0
        NM23=0
        SEG=SEGM*WV
C SET EVEN SIDE INDICATOR
        EVSD=.TRUE.
        IF(NC.NE.2*(NC/2))EVSD=.FALSE.
        IF(IP.NE.0)GOTO 4
        NDNPLT(NP)=0
        IF(NP.NE.1)NDNPLT(NP)=NDNPLT(NP-1)
        RETURN
C SET REFERENCE DIRECTION FOR INTERNAL ANGLE CHECK
4       IC=(NC+1)/2
100     DA=SQRT((PC(1,IC)-PC(1,1))**2+(PC(2,IC)-PC(2,1))**2
     &+(PC(3,IC)-PC(3,1))**2)
        DB=SQRT((PC(1,2)-PC(1,1))**2+(PC(2,2)-PC(2,1))**2
     &+(PC(3,2)-PC(3,1))**2)
        DO 5 IQ=1,3
        CA(IQ)=(PC(IQ,IC)-PC(IQ,1))/DA
```

```
5       CB(IQ)=(PC(IQ,2)-PC(IQ,1))/DB
        CC=CA(1)*CB(1)+CA(2)*CB(2)+CA(3)*CB(3)
        SS=SQRT(1.-CC*CC)
        IF(SS.GE.0.02)GOTO 105
        IC=IC+1
        GOTO 100
105     ISD=0
        IPD=0
        CG(1)=(CB(2)*CA(3)-CA(2)*CB(3))/SS
        CG(2)=(CB(3)*CA(1)-CA(3)*CB(1))/SS
        CG(3)=(CB(1)*CA(2)-CA(1)*CB(2))/SS
C CHECK FOR INTERNAL ANGLES > 180 DEGREES; FIND MAXIMUM LENGTH SIDE
        DO 10 IC0=1,NC
        IC=IC0+1
        IF(IC0.EQ.NC)IC=1
        IC2=IC+1
        IF(IC.EQ.NC)IC2=1
        DB=SQRT((PC(1,IC2)-PC(1,IC))**2+(PC(2,IC2)-PC(2,IC))**2
     &+(PC(3,IC2)-PC(3,IC))**2)
        IF(DB.LE.DMX)GOTO 15
        DMX=DB
        ISMX=IC
15      DO 11 IQ=1,3
        CA(IQ)=-CB(IQ)
11      CB(IQ)=(PC(IQ,IC2)-PC(IQ,IC))/DB
        CC=CA(1)*CB(1)+CA(2)*CB(2)+CA(3)*CB(3)
        SS=SQRT(1.-CC*CC)
        IF(SS.GE.0.02)GOTO 110
C INTERNAL ANGLE = 180 DEGREES
        IPD=IPD+1
        GOTO 10
110     CD(1)=(CB(2)*CA(3)-CA(2)*CB(3))/SS
        CD(2)=(CB(3)*CA(1)-CA(3)*CB(1))/SS
        CD(3)=(CB(1)*CA(2)-CA(1)*CB(2))/SS
        DP=CG(1)*CD(1)+CG(2)*CD(2)+CG(3)*CD(3)
        IF(DP.GT.0.)ISD=ISD+1
        IF(DP.LT.0.)IDC=IC
10      CONTINUE
        IF(ISD+IPD.EQ.NC)GOTO 20
        IF(ISD.EQ.1 .OR. NC-ISD-IPD.EQ.1)GOTO 95
        WRITE(6,500)NP
500     FORMAT(' ***** ERROR *****  PLATE',I3,' HAS MORE THAN ONE ',
     &'INTERNAL ANGLE GREATER THAN 180 DEGREES')
        NDNPLT(NP)=0
        IF(NP.NE.1)NDNPLT(NP)=NDNPLT(NP-1)
        IOK=0
        RETURN
20      CONTINUE
C NO INTERNAL ANGLES > 180 DEGREES; SET ISMX = GENERATING SIDE
        IF(NC.EQ.4 .AND. IP.EQ.1)ISMX=1
        IF(NC.EQ.4 .AND. IP.EQ.2)ISMX=2
        IF(IGS.NE.0)ISMX=IGS
        IC2=ISMX
        IC=ISMX+1
        IF(ISMX.EQ.NC)IC=1
        IIS=0
        GOTO 200
95      CONTINUE
```

```
C INTERNAL ANGLE AT CORNER IDC > 180 DEGREES; SET GENERATING SIDE
      IF(ISD.NE.1)GOTO 97
      DO 98 IQ=1,3
98    CG(IQ)=-CG(IQ)
97    IC2=IDC+1
      IF(IDC.EQ.NC)IC2=1
      IF(ISD.EQ.1)IC2=2
      IC=IC2+1
      IF(IC2.EQ.NC)IC=1
      IIS=0
C SET GRID POINTS STARTING AT ENDPOINTS OF GENERATING SIDE
200   IF(EVSD)MDM=MDME
      IF(.NOT.EVSD)MDM=MDMO
C INCREMENT SIDE 2; CHECK FOR COMPLETION
      IC2=IC2+1
      IF(IC2.GT.NC)IC2=1
      IF(IC2.EQ.IC .AND. IIS.NE.0)GOTO 205
      IIS=IIS+1
C INCREMENT SIDE 1
      IC=IC-1
      IF(IC.EQ.0)IC=NC
      IF(IC.NE.IC2)GOTO 210
C REMAINING PIECE IS TRIANGLE; PUT EXTRA CORNER AT MIDPOINT OF
C  LONGEST REMAINING SIDE
      DA=SQRT((PC(1,IC)-GPT(IIS-1,1,1))**2+(PC(2,IC)
    &-GPT(IIS-1,1,2))**2+(PC(3,IC)-GPT(IIS-1,1,3))**2)
      DB=SQRT((PC(1,IC2)-GPT(IIS-1,2,1))**2+(PC(2,IC2)
    &-GPT(IIS-1,2,2))**2+(PC(3,IC2)-GPT(IIS-1,2,3))**2)
      IF(DA.GT.DB)GOTO 215
      DO 220 IQ=1,3
      GPT(IIS,1,IQ)=PC(IQ,IC)
220   GPT(IIS,2,IQ)=(PC(IQ,IC2)+GPT(IIS-1,2,IQ))/2.
      IC2=IC2-1
      IF(IC.EQ.0)IC=NC
      GOTO 225
215   DO 230 IQ=1,3
      GPT(IIS,1,IQ)=(PC(IQ,IC)+GPT(IIS-1,1,IQ))/2.
230   GPT(IIS,2,IQ)=PC(IQ,IC2)
      IC2=IC2-1
      IF(IC.EQ.0)IC=NC
      GOTO 225
C REMAINING PIECE IS AT MINIMUM A QUADRALATERAL
210   DO 235 IQ=1,3
      GPT(IIS,1,IQ)=PC(IQ,IC)
235   GPT(IIS,2,IQ)=PC(IQ,IC2)
225   IF(IIS.EQ.1)GOTO 200
C CHECK SIDES FOR EXTRAS NEEDED GRID POINT
      DA=SQRT((GPT(IIS,1,1)-GPT(IIS-1,1,1))**2+
    &(GPT(IIS,1,2)-GPT(IIS-1,1,2))**2+(GPT(IIS,1,3)-GPT(IIS-1,1,3))**2)
      DB=SQRT((GPT(IIS,2,1)-GPT(IIS-1,2,1))**2+
    &(GPT(IIS,2,2)-GPT(IIS-1,2,2))**2+(GPT(IIS,2,3)-GPT(IIS-1,2,3))**2)
      NAS=DA/SEG+0.99
      NBS=DB/SEG+0.99
      NES=MAX0(NAS,NBS)
      IF(NC.EQ.4 .AND. IP.EQ.3)NES=MAX0(2,NES)
      IF(NES.EQ.1)GOTO 200
C SET DIRECTIONAL COSINES FOR SIDES
      DO 115 IQ=1,3
```

```
        CA(IQ)=(GPT(IIS,1,IQ)-GPT(IIS-1,1,IQ))/DA
115     CB(IQ)=(GPT(IIS,2,IQ)-GPT(IIS-1,2,IQ))/DB
C CHECK NUMBER OF MODES AGAINST MAXIMUM MODE DIFFERENCE PARAMETER
        IF(NAS-NBS.LT.MDM)GOTO 120
C SHORTEN SIDE 1; TOGGLE EVSD
        DO 125 IQ=1,3
125     GPT(IIS,1,IQ)=GPT(IIS-1,1,IQ)+DB*CA(IQ)
        DA=DB
        NAS=NBS
        NES=NBS
        IC=IC+1
        IF(IC.GT.NC)IC=1
        EVSD=.NOT.EVSD
        GOTO 130
120     IF(NBS-NAS.LT.MDM)GOTO 130
C SHORTEN SIDE 2; TOGGLE EVSD
        DO 135 IQ=1,3
135     GPT(IIS,2,IQ)=GPT(IIS-1,2,IQ)+DA*CB(IQ)
        DB=DA
        NBS=NAS
        NES=NAS
        IC2=IC2-1
        IF(IC2.EQ.0)IC2=NC
        EVSD=.NOT.EVSD
130     IF(NES.EQ.1)GOTO 200
C CHECK FOR MODES WITH INTERNAL ANGLES > 180 DEGREES
        DD=SQRT((GPT(IIS,1,1)-GPT(IIS,2,1))**2+
     &(GPT(IIS,1,2)-GPT(IIS,2,2))**2+(GPT(IIS,1,3)-GPT(IIS,2,3))**2)
        DO 240 IQ=1,3
        CD(IQ)=(GPT(IIS,1,IQ)-GPT(IIS,2,IQ))/DD
        GPT(IIS+NES-1,1,IQ)=GPT(IIS,1,IQ)
240     GPT(IIS+NES-1,2,IQ)=GPT(IIS,2,IQ)
        CC=CA(1)*CD(1)+CA(2)*CD(2)+CA(3)*CD(3)
        SS=SQRT(1.-CC*CC)
        IF(SS.LT.0.02)GOTO 244
        CE(1)=(CA(2)*CD(3)-CA(3)*CD(2))/SS
        CE(2)=(CA(3)*CD(1)-CA(1)*CD(3))/SS
        CE(3)=(CA(1)*CD(2)-CA(2)*CD(1))/SS
        DP=CG(1)*CE(1)+CG(2)*CE(2)+CG(3)*CE(3)
        IF(DP.GT.0.)GOTO 244
        WRITE(6,501)NP
501     FORMAT(' ***** ERROR *****  PLATE',I3,' HAS MODES WITH ',
     &'INTERNAL ANGLES GREATER THAN 180 DEGREES')
244     CONTINUE
C ADD EXTRA NEEDED GRID POINTS
        DO 245 IR=1,NES
        DO 245 IQ=1,3
        GPT(IIS+IR-1,1,IQ)=GPT(IIS-1,1,IQ)+IR*DA/NES*CA(IQ)
        GPT(IIS+IR-1,2,IQ)=GPT(IIS-1,2,IQ)+IR*DB/NES*CB(IQ)
245     CONTINUE
        IIS=IIS+NES-1
        GOTO 200
C FIND LONGEST DISTANCE BETWEEN GRID POINTS
205     DMX=0.
        DO 70 IS=1,IIS
        DA=SQRT((GPT(IS,2,1)-GPT(IS,1,1))**2+(GPT(IS,2,2)-GPT(IS,1,2))**2
     &+(GPT(IS,2,3)-GPT(IS,1,3))**2)
        IF(DA.GT.DMX)DMX=DA
```

```
70      CONTINUE
C SET NUMBER OF MODES ALONG AND ORTHONAL TO GENERATING SIDE
        NMP=IIS-1
        NMO=DMX/SEG+0.99
        NMO=MAX0(2,NMO)
        IF(IRE.EQ.1 .AND. (ISMX.EQ.1 .OR. ISMX.EQ.3))NM12=NMO
        IF(IRE.EQ.1 .AND. (ISMX.EQ.1 .OR. ISMX.EQ.3))NM23=NMP
        IF(IRE.EQ.1 .AND. (ISMX.EQ.2 .OR. ISMX.EQ.4))NM12=NMP
        IF(IRE.EQ.1 .AND. (ISMX.EQ.2 .OR. ISMX.EQ.4))NM23=NMO
        NDN=0
        IF(NP.NE.1)NDN=NDNPLT(NP-1)
        MPL1=NMP*(NMO-1)
        MPL2=MPL1+NMO*(NMP-1)
C GENERATE MODES IN DIRECTION ALONG GENERATING SIDE
        DO 75 I=1,NMP
        DA=SQRT((GPT(I,2,1)-GPT(I,1,1))**2+(GPT(I,2,2)-GPT(I,1,2))**2
     &+(GPT(I,2,3)-GPT(I,1,3))**2)
        DB=SQRT((GPT(I+1,2,1)-GPT(I+1,1,1))**2+(GPT(I+1,2,2)-GPT(I+1,1,2))**2
     &+(GPT(I+1,2,3)-GPT(I+1,1,3))**2)
        DO 76 IQ=1,3
        CA(IQ)=(GPT(I,2,IQ)-GPT(I,1,IQ))/DA
76      CB(IQ)=(GPT(I+1,2,IQ)-GPT(I+1,1,IQ))/DB
        DDA=DA/NMO
        DDB=DB/NMO
        DO 80 J=1,NMO-1
        NDN=NDN+1
        IF(IRE.EQ.1)IQUAD=-3
        IF(IRE.EQ.0)IQUAD=0
        DO 80 IQ=1,3
        PA(NDN,1,IQ)=GPT(I,1,IQ)+J*DDA*CA(IQ)
        PA(NDN,2,IQ)=GPT(I,1,IQ)+(J-1)*DDA*CA(IQ)
        PA(NDN,3,IQ)=GPT(I+1,1,IQ)+(J-1)*DDB*CB(IQ)
        PA(NDN,4,IQ)=GPT(I+1,1,IQ)+J*DDB*CB(IQ)
        PB(NDN,1.IQ)=PA(NDN,1.IQ)
        PB(NDN,2,IQ)=GPT(I,1,IQ)+(J+1)*DDA*CA(IQ)
        PB(NDN,3,IQ)=GPT(I+1,1,IQ)+(J+1)*DDB*CB(IQ)
        PB(NDN,4,IQ)=PA(NDN,4,IQ)
80      CONTINUE
75      CONTINUE
        NDNPLT(NP)=NDN
        IF(NC.EQ.4 .AND. IP.NE.3)RETURN
C GENERATE MODES ORTHOGONAL TO GENERATING SIDE
        DO 85 I=1,NMO
        DO 90 J=1,NMP-1
        NDN=NDN+1
        IF(IRE.EQ.1)IQUAD=-3
        IF(IRE.EQ.0)IQUAD=0
        DA=SQRT((GPT(J,2,1)-GPT(J,1,1))**2+(GPT(J,2,2)
     &-GPT(J,1,2))**2+(GPT(J,2,3)-GPT(J,1,3))**2)
        DB=SQRT((GPT(J+1,2,1)-GPT(J+1,1,1))**2+(GPT(J+1,2,2)
     &-GPT(J+1,1,2))**2+(GPT(J+1,2.3)-GPT(J+1,1,3))**2)
        DG=SQRT((GPT(J+2,2,1)-GPT(J+2,1,1))**2+(GPT(J+2,2,2)
     &-GPT(J+2,1,2))**2+(GPT(J+2,2.3)-GPT(J+2,1,3))**2)
        DDA=DA/NMO
        DDB=DB/NMO
        DDG=DG/NMO
        DO 90 IQ=1,3
        CA(IQ)=(GPT(J,2,IQ)-GPT(J,1,IQ))/DA
```

```
      CB(IQ)=(GPT(J+1,2,IQ)-GPT(J+1,1,IQ))/DB
      CG(IQ)=(GPT(J+2,2,IQ)-GPT(J+2,1,IQ))/DG
      PA(NDN,1,IQ)=GPT(J+1,1,IQ)+(I-1)*DDB*CB(IQ)
      PA(NDN,2,IQ)=GPT(J,1,IQ)+(I-1)*DDA*CA(IQ)
      PA(NDN,3,IQ)=GPT(J,1,IQ)+I*DDA*CA(IQ)
      PA(NDN,4,IQ)=GPT(J+1,1,IQ)+I*DDB*CB(IQ)
      PB(NDN.1,IQ)=PA(NDN.1.IQ)
      PB(NDN,2,IQ)=GPT(J+2,1.IQ)+(I-1)*DDG*CG(IQ)
      PB(NDN,3,IQ)=GPT(J+2,1,IQ)+I*DDG*CG(IQ)
      PB(NDN,4,IQ)=PA(NDN,4,IQ)
90    CONTINUE
85    CONTINUE
      NDNPLT(NP)=NDN
      RETURN
      END
```

APPENDIX 8

SUBROUTINE POPLOV

```
      SUBROUTINE POPLOV(NPLTS,PCN,NCNRS,TOUCH
     2,SEGM,PA,PB,NOVT,NPLTM,IPL,IPLM,ICN
     3,IOVT,DOVL,ITK,NOPL,IQUAD,WV,NDNPLT,OVEP)
      DIMENSION PCN(3,ICN,IPL),NCNRS(1),SEGM(1)
     2,PA(IPLM,4,3),PB(IPLM,4,3),IOVT(IPLM,4),DOVL(1)
     3,ITK(1),IQUAD(1),NDNPLT(1),OVEP(IPLM,3,2)
      DIMENSION IBC(2),OE(3,2),CE(3)
     2,CA(3),CB(3),CA1(3),CA2(3),CB1(3),CB2(3),OMSP(3,4,2)
      DIMENSION OEI(3,2),OEJ(3,2),OEK(3,2),VDM(3)
      IF(NPLTS.LT.2)RETURN
      TPSI=0.258819
      SPSI=SQRT(1.-TPSI**2)
      CLEN=WV/25.
      NOPL=0
      NOVT=0
C   CHECK FOR TOUCHING PLATES
      DO 100 NPA=1,NPLTS-1
      DO 100 NPB=NPA+1,NPLTS
      DO 110 ISA=1,NCNRS(NPA)
C   COMPUTE DIRECTIONAL COSINES OF SIDE ISA, PLATE NPA
      ISA1=ISA+1
      IF(ISA.EQ.NCNRS(NPA))ISA1=1
      CXA=PCN(1,ISA1,NPA)-PCN(1,ISA,NPA)
      CYA=PCN(2,ISA1,NPA)-PCN(2,ISA,NPA)
      CZA=PCN(3,ISA1,NPA)-PCN(3,ISA,NPA)
      DA=SQRT(CXA*CXA+CYA*CYA+CZA*CZA)
      CXA=CXA/DA
      CYA=CYA/DA
      CZA=CZA/DA
      ITCH=0
      DO 115 ICB=1,NCNRS(NPB)
C   COMPUTE DISTANCE BETWEEN CORNER ICB, PLATE NPB AND
C     SIDE ISA, PLATE NPA
      XAB=PCN(1,ICB,NPB)-PCN(1,ISA,NPA)
      YAB=PCN(2,ICB,NPB)-PCN(2,ISA,NPA)
      ZAB=PCN(3,ICB,NPB)-PCN(3,ISA,NPA)
      DSQ=XAB*XAB+YAB*YAB+ZAB*ZAB-(XAB*CXA+YAB*CYA+ZAB*CZA)**2
      IF(DSQ.GT.TOUCH**2)GO TO 115
      ITCH=ITCH+1
      IBC(ITCH)=ICB
```

```
115     CONTINUE
        IF(ITCH.NE.2)GOTO 110
C   CORNERS OF PLATE NPB TOUCHES SIDE ISA OF PLATE NPA
C     FIND ENDPOINTS OF OVERLAP SEGMENT
        ISB=IBC(1)
        IF(IBC(1).EQ.1 .AND. IBC(2).EQ.NCNRS(NPB))ISB=NCNRS(NPB)
        IOV=0
        DO 120 J=1,2
        ICB=IBC(J)
        DCJL=(PCN(1,ISA,NPA)-PCN(1,ICB,NPB))**2
     2+(PCN(2,ISA,NPA)-PCN(2,ICB,NPB))**2
     3+(PCN(3,ISA,NPA)-PCN(3,ICB,NPB))**2
        DCJH=(PCN(1,ISA1,NPA)-PCN(1,ICB,NPB))**2
     2+(PCN(2,ISA1,NPA)-PCN(2,ICB,NPB))**2
     3+(PCN(3,ISA1,NPA)-PCN(3,ICB,NPB))**2
        IF(DCJL+DCJH-DA*DA.GT.2.*TOUCH**2)GOTO 120
        IOV=IOV+1
        DO 130 IQ=1,3
130     OE(IQ,IOV)=PCN(IQ,ICB,NPB)
120     CONTINUE
        IF(IOV.EQ.2)GOTO 140
        ICB=IBC(1)
        ICB1=IBC(2)
        DBS=(PCN(1,ICB,NPB)-PCN(1,ICB1,NPB))**2
     2+(PCN(2,ICB,NPB)-PCN(2,ICB1,NPB))**2
     3+(PCN(3,ICB,NPB)-PCN(3,ICB1,NPB))**2
        DO 150 J=1,2
        ICA=ISA
        IF(J.EQ.2)ICA=ISA1
        DCJL=(PCN(1,ICB,NPB)-PCN(1,ICA,NPA))**2
     2+(PCN(2,ICB,NPB)-PCN(2,ICA,NPA))**2
     3+(PCN(3,ICB,NPB)-PCN(3,ICA,NPA))**2
        DCJH=(PCN(1,ICB1,NPB)-PCN(1,ICA,NPA))**2
     2+(PCN(2,ICB1,NPB)-PCN(2,ICA,NPA))**2
     3+(PCN(3,ICB1,NPB)-PCN(3,ICA,NPA))**2
        IF(DCJL+DCJH-DBS.GT.2.*TOUCH**2)GOTO 150
        IOV=IOV+1
        DO 160 IQ=1,3
160     OE(IQ,IOV)=PCN(IQ,ICA,NPA)
        IF(IOV.NE.2)GOTO 150
        DOV=SQRT((OE(1,1)-OE(1,2))**2+(OE(2,1)-OE(2,2))**2
     2+(OE(3,1)-OE(3,2))**2)
        IF(DOV.GT.TOUCH)GOTO 165
        IOV=1
150     CONTINUE
        IF(IOV.NE.2)GOTO 110
140     DOV=SQRT((OE(1,1)-OE(1,2))**2+(OE(2,1)-OE(2,2))**2
     2+(OE(3,1)-OE(3,2))**2)
        IF(DOV.GT.TOUCH)GOTO 165
110     CONTINUE
        GOTO 100
C   WRITE OVERLAB MODES INTO TABLE
165     NOPL=NOPL+1
        IOVT(NOPL,1)=NPA
        IOVT(NOPL,2)=ISA
        IOVT(NOPL,3)=NPB
        IOVT(NOPL,4)=ISB
        DOVL(NOPL)=DOV
```

```
      DO 166 IQ=1,3
      OVEP(NOPL,IQ,1)=OE(IQ,1)
166   OVEP(NOPL,IQ,2)=OE(IQ,2)
      CALL FGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCNRS,PCN,ICN,IPL,SEGM
     &,NDNPLT,PA,PB,IPLM,WV,TOUCH.TPSI,SPSI
     &,OMSP,ITK(NOPL))
100   CONTINUE
      IF(NOPL.EQ.0)RETURN
      IF(NOPL.LE.2)GOTO 200
C   SEARCH OVERLAP TABLE, REMOVING UNNECESSARY MODES
      DO 170  I=1,NOPL-2
      NPAI=IOVT(I,1)
      ISAI=IOVT(I,2)
      NPBI=IOVT(I,3)
      ISBI=IOVT(I,4)
      DOVI=DOVL(I)
      DDI=DOVI/IABS(ITK(I))
      DO 171 IQ=1,3
      OEI(IQ,1)=OVEP(I,IQ,1)
      OEI(IQ,2)=OVEP(I,IQ,2)
171   CE(IQ)=(OEI(IQ,2)-OEI(IQ,1))/DOVI
      DO 175 J=I+1,NOPL-1
      NPAJ=IOVT(J,1)
      ISAJ=IOVT(J,2)
      NPBJ=IOVT(J,3)
      ISBJ=IOVT(J,4)
      IF(NPAI.NE.NPAJ .OR. ISAI.NE.ISAJ)GOTO 175
      DOVJ=DOVL(J)
      DDJ=DOVJ/IABS(ITK(J))
      DO 176 IQ=1,3
      OEJ(IQ,1)=OVEP(J,IQ,1)
      OEJ(IQ,2)=OVEP(J,IQ,2)
176   VDM(IQ)=(OEJ(IQ,2)-OEJ(IQ,1))/DOVJ
      IF(CE(1)*VDM(1)+CE(2)*VDM(2)+CE(3)*VDM(3).GT.0.)GOTO 177
      DO 178 IQ=1,3
      OE(IQ,1)=OEJ(IQ,2)
      OEJ(IQ,2)=OEJ(IQ,1)
178   OEJ(IQ,1)=OE(IQ,1)
177   CONTINUE
      DO 180 K=J+1,NOPL
      NPAK=IOVT(K,1)
      ISAK=IOVT(K,2)
      NPBK=IOVT(K,3)
      ISBK=IOVT(K,4)
      IF(NPBI.NE.NPAK .OR. ISBI.NE.ISAK .OR. NPBJ.NE.NPBK .OR.
     2ISBJ.NE.ISBK)GOTO 180
      DOVK=DOVL(K)
      DDK=DOVK/IABS(ITK(K))
      DO 181 IQ=1,3
      OEK(IQ,1)=OVEP(K,IQ,1)
      OEK(IQ,2)=OVEP(K,IQ,2)
181   VDM(IQ)=(OEK(IQ,2)-OEK(IQ,1))/DOVK
      IF(CE(1)*VDM(1)+CE(2)*VDM(2)+CE(3)*VDM(3).GT.0.)GOTO 182
      DO 183 IQ=1,3
      OE(IQ,1)=OEK(IQ,2)
      OEK(IQ,2)=OEK(IQ,1)
183   OEK(IQ,1)=OE(IQ,1)
182   CONTINUE
```

```
      IF(ABS(DOVI-DOVJ).GT.CLEN .OR. ABS(DOVI-DOVK).GT.CLEN .OR.
    &ABS(DOVJ-DOVK).GT.CLEN)GOTO 400
      ICK=0
      IF(IABS(ITK(I)).EQ.IABS(ITK(J)))ICK=1
      IF(IABS(ITK(I)).EQ.IABS(ITK(K)))ICK=ICK+2
      IF(IABS(ITK(J)).EQ.IABS(ITK(K)))ICK=ICK+4
      IF(ICK.NE.7)GOTO 405
      IF(ITK(K).LT.0)GOTO 410
      ITK(K)=-ITK(K)
      GOTO 175
410   ITK(J)=-IABS(ITK(J))
      GOTO 175
405   IF(ICK.EQ.1)ITK(K)=-IABS(ITK(K))
      IF(ICK.EQ.2)ITK(J)=-IABS(ITK(J))
      IF(ICK.EQ.4)ITK(I)=-IABS(ITK(I))
      GOTO 175
400   ICK=0
      IF(ABS(DDI-DDJ).LE.CLEN/MAX0(IABS(ITK(I)),IABS(ITK(J))))ICK=1
      IF(ABS(DDI-DDK).LE.CLEN/MAX0(IABS(ITK(I)),IABS(ITK(K))))
    &ICK=ICK+2
      IF(ABS(DDJ-DDK).LE.CLEN/MAX0(IABS(ITK(J)),IABS(ITK(K))))
    &ICK=ICK+4
      IF(ICK.LT.1)GOTO 175
      IF(ICK.EQ.7)GOTO 415
      DOV=AMIN1(DOVK,DOVJ,DOVI)
      IF(ICK.EQ.4 .AND. ABS(DOVI-DOV).GT.CLEN)GOTO 175
      IF(ICK.EQ.2 .AND. ABS(DOVJ-DOV).GT.CLEN)GOTO 175
      IF(ICK.EQ.1 .AND. ABS(DOVK-DOV).GT.CLEN)GOTO 175
C    CHECK GROUP I, PLATE A, ENDPOINT 1 = GROUP J, PLATE A, ENDPOINT 1
415   DIJ=SQRT((OEI(1,1)-OEJ(1,1))**2+(OEI(2,1)-OEJ(2,1))**2
    &+(OEI(3,1)-OEJ(3,1))**2)
      IF(DIJ.LE.CLEN)GOTO 440
      DO 420 IQ=1,3
420   VDM(IQ)=(OEI(IQ,1)-OEJ(IQ,1))/DIJ
      IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).LT.0.)GOTO 425
      NN=DIJ/DDJ+0.5
      DO 430 IQ=1,3
430   VDM(IQ)=OEJ(IQ,1)+NN*DDJ*CE(IQ)
      DIJ=(OEI(1,1)-VDM(1))**2+(OEI(2,1)-VDM(2))**2
    &+(OEI(3,1)-VDM(3))**2
      IF(DIJ.LT.CLEN**2)GOTO 440
      GOTO 175
425   NN=DIJ/DDI+0.5
      DO 435 IQ=1,3
435   VDM(IQ)=OEI(IQ,1)+NN*DDI*CE(IQ)
      DIJ=(OEJ(1,1)-VDM(1))**2+(OEJ(2,1)-VDM(2))**2
    &+(OEJ(3,1)-VDM(3))**2
      IF(DIJ.LT.CLEN**2)GOTO 440
      GOTO 175
C    CHECK GROUP I, PLATE A, ENDPOINT 2 = GROUP J, PLATE A, ENDPOINT 2
440   DIJ=SQRT((OEI(1,2)-OEJ(1,2))**2+(OEI(2,2)-OEJ(2,2))**2
    &+(OEI(3,2)-OEJ(3,2))**2)
      IF(DIJ.LE.CLEN)GOTO 465
      DO 445 IQ=1,3
445   VDM(IQ)=(OEI(IQ,2)-OEJ(IQ,2))/DIJ
      IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).GT.0.)GOTO 450
      NN=DIJ/DDJ+0.5
      DO 455 IQ=1,3
```

```
455     VDM(IQ)=OEJ(IQ,2)-NN*DDJ*CE(IQ)
        DIJ=(OEI(1,2)-VDM(1))**2+(OEI(2,2)-VDM(2))**2
     &+(OEI(3,2)-VDM(3))**2
        IF(DIJ.LT.CLEN**2)GOTO 465
        GOTO 175
450     NN=DIJ/DDI+0.5
        DO 460 IQ=1,3
460     VDM(IQ)=OEI(IQ,2)-NN*DDI*CE(IQ)
        DIJ=(OEJ(1,2)-VDM(1))**2+(OEJ(2,2)-VDM(2))**2
     &+(OEJ(3,2)-VDM(3))**2
        IF(DIJ.LT.CLEN**2)GOTO 465
        GOTO 175
C      CHECK GROUP I, PLATE B, ENDPOINT 1 = GROUP K, PLATE A, ENDPOINT 1
465     DIJ=SQRT((OEI(1,1)-OEK(1,1))**2+(OEI(2,1)-OEK(2,1))**2
     &+(OEI(3,1)-OEK(3,1))**2)
        IF(DIJ.LE.CLEN)GOTO 490
        DO 470 IQ=1,3
470     VDM(IQ)=(OEI(IQ,1)-OEK(IQ,1))/DIJ
        IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).LT.0.)GOTO 475
        NN=DIJ/DDK+0.5
        DO 480 IQ=1,3
480     VDM(IQ)=OEK(IQ,1)+NN*DDK*CE(IQ)
        DIJ=(OEI(1,1)-VDM(1))**2+(OEI(2,1)-VDM(2))**2
     &+(OEI(3,1)-VDM(3))**2
        IF(DIJ.LT.CLEN**2)GOTO 490
        GOTO 175
475     NN=DIJ/DDI+0.5
        DO 485 IQ=1,3
485     VDM(IQ)=OEI(IQ,1)+NN*DDI*CE(IQ)
        DIJ=(OEK(1,1)-VDM(1))**2+(OEK(2,1)-VDM(2))**2
     &+(OEK(3,1)-VDM(3))**2
        IF(DIJ.LT.CLEN**2)GOTO 490
        GOTO 175
C      CHECK GROUP I, PLATE B, ENDPOINT 2 = GROUP K, PLATE A, ENDPOINT 2
490     DIJ=SQRT((OEI(1,2)-OEK(1,2))**2+(OEI(2,2)-OEK(2,2))**2
     &+(OEI(3,2)-OEK(3,2))**2)
        IF(DIJ.LE.CLEN)GOTO 515
        DO 495 IQ=1,3
495     VDM(IQ)=(OEI(IQ,2)-OEK(IQ,2))/DIJ
        IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).GT.0.)GOTO 500
        NN=DIJ/DDK+0.5
        DO 505 IQ=1,3
505     VDM(IQ)=OEK(IQ,2)-NN*DDK*CE(IQ)
        DIJ=(OEI(1,2)-VDM(1))**2+(OEI(2,2)-VDM(2))**2
     &+(OEI(3,2)-VDM(3))**2
        IF(DIJ.LT.CLEN**2)GOTO 515
        GOTO 175
500     NN=DIJ/DDI+0.5
        DO 510 IQ=1,3
510     VDM(IQ)=OEI(IQ,2)-NN*DDI*CE(IQ)
        DIJ=(OEK(1,2)-VDM(1))**2+(OEK(2,2)-VDM(2))**2
     &+(OEK(3,2)-VDM(3))**2
        IF(DIJ.LT.CLEN**2)GOTO 515
        GOTO 175
C      CHECK GROUP J, PLATE B, ENDPOINT 1 = GROUP K, PLATE B, ENDPOINT 1
515     DIJ=SQRT((OEJ(1,1)-OEK(1,1))**2+(OEJ(2,1)-OEK(2,1))**2
     &+(OEJ(3,1)-OEK(3,1))**2)
        IF(DIJ.LE.CLEN)GOTO 540
```

```
      DO 520 IQ=1,3
520   VDM(IQ)=(OEJ(IQ,1)-OEK(IQ,1))/DIJ
      IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).LT.0.)GOTO 525
      NN=DIJ/DDK+0.5
      DO 530 IQ=1,3
530   VDM(IQ)=OEK(IQ,1)+NN*DDK*CE(IQ)
      DIJ=(OEJ(1,1)-VDM(1))**2+(OEJ(2,1)-VDM(2))**2
    &+(OEJ(3,1)-VDM(3))**2
      IF(DIJ.LT.CLEN**2)GOTO 540
      GOTO 175
525   NN=DIJ/DDJ+0.5
      DO 535 IQ=1,3
535   VDM(IQ)=OEJ(IQ,1)+NN*DDJ*CE(IQ)
      DIJ=(OEK(1,1)-VDM(1))**2+(OEK(2,1)-VDM(2))**2
    &+(OEK(3,1)-VDM(3))**2
      IF(DIJ.LT.CLEN**2)GOTO 540
      GOTO 175
C    CHECK GROUP J, PLATE B, ENDPOINT 2 = GROUP K, PLATE B, ENDPOINT 2
540   DIJ=SQRT((OEJ(1,2)-OEK(1,2))**2+(OEJ(2,2)-OEK(2,2))**2
    &+(OEJ(3,2)-OEK(3,2))**2)
      IF(DIJ.LE.CLEN)GOTO 565
      DO 545 IQ=1,3
545   VDM(IQ)=(OEJ(IQ,2)-OEK(IQ,2))/DIJ
      IF(VDM(1)*CE(1)+VDM(2)*CE(2)+VDM(3)*CE(3).GT.0.)GOTO 550
      NN=DIJ/DDK+0.5
      DO 555 IQ=1,3
555   VDM(IQ)=OEK(IQ,2)-NN*DDK*CE(IQ)
      DIJ=(OEJ(1,2)-VDM(1))**2+(OEJ(2,2)-VDM(2))**2
    &+(OEJ(3,2)-VDM(3))**2
      IF(DIJ.LT.CLEN**2)GOTO 565
      GOTO 175
550   NN=DIJ/DDJ+0.5
      DO 560 IQ=1,3
560   VDM(IQ)=OEJ(IQ,2)-NN*DDJ*CE(IQ)
      DIJ=(OEK(1,2)-VDM(1))**2+(OEK(2,2)-VDM(2))**2
    &+(OEK(3,2)-VDM(3))**2
      IF(DIJ.LT.CLEN**2)GOTO 565
      GOTO 175
565   IF(ICK.EQ.7 .OR. ICK.EQ.1)ITK(K)=-IABS(ITK(K))
      IF(ICK.EQ.2)ITK(J)=-IABS(ITK(J))
      IF(ICK.EQ.4)ITK(I)=-IABS(ITK(I))
      GOTO 175
180   CONTINUE
175   CONTINUE
170   CONTINUE
C  CONSTRUCT OVERLAP MODES
200   DO 205 IV=1,NOPL
      IF(ITK(IV).GT.0)GOTO 201
      ITK(IV)=0
      GOTO 205
201   NPA=IOVT(IV,1)
      ISA=IOVT(IV,2)
      NPB=IOVT(IV,3)
      ISB=IOVT(IV,4)
      DOV=DOVL(IV)
      DO 206 IQ=1,3
      OE(IQ,1)=OVEP(IV,IQ,1)
      OE(IQ,2)=OVEP(IV,IQ,2)
```

```
206       CE(IQ)=(OE(IQ,2)-OE(IQ,1))/DOV
          CALL FGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCNRS,PCN,ICN,IPL,SEGM
     &,NDNPLT,PA,PB,IPLM,WV,TOUCH,TPSI,SPSI
     &,OMSP,NOV)
          DA=SQRT((OMSP(1,3,1)-OMSP(1,4,1))**2+(OMSP(2,3,1)-OMSP(2,4,1))
     &**2+(OMSP(3,3,1)-OMSP(3,4,1))**2)
          DB=SQRT((OMSP(1,3,2)-OMSP(1,4,2))**2+(OMSP(2,3,2)-OMSP(2,4,2))
     &**2+(OMSP(3,3,2)-OMSP(3,4,2))**2)
          DSA=DA/NOV
          DSB=DB/NOV
          DSE=DOV/NOV
          DO 285 IQ=1,3
          CA(IQ)=(OMSP(IQ,3,1)-OMSP(IQ,4,1))/DA
285       CB(IQ)=(OMSP(IQ,3,2)-OMSP(IQ,4,2))/DB
          DO 281 IQ=1,3
          CA1(IQ)=OMSP(IQ,4,1)-OMSP(IQ,1,1)
          CA2(IQ)=OMSP(IQ,3,1)-OMSP(IQ,2,1)
          CB1(IQ)=OMSP(IQ,4,2)-OMSP(IQ,1,2)
281       CB2(IQ)=OMSP(IQ,3,2)-OMSP(IQ,2,2)
          DA1=SQRT(CA1(1)**2+CA1(2)**2+CA1(3)**2)
          DA2=SQRT(CA2(1)**2+CA2(2)**2+CA2(3)**2)
          DB1=SQRT(CB1(1)**2+CB1(2)**2+CB1(3)**2)
          DB2=SQRT(CB2(1)**2+CB2(2)**2+CB2(3)**2)
          TA1=CE(1)*CA1(1)+CE(2)*CA1(2)+CE(3)*CA1(3)
          TA2=-(CE(1)*CA2(1)+CE(2)*CA2(2)+CE(3)*CA2(3))
          TB1=(CE(1)*CB1(1)+CE(2)*CB1(2)+CE(3)*CB1(3))
          TB2=-(CE(1)*CB2(1)+CE(2)*CB2(2)+CE(3)*CB2(3))
C    FILL IN OVERLAP MODES
          DO 330 I=1,NOV
          DO 335 IQ=1,3
          NN=NPLTM+NOVT+I
          PA(NN,1,IQ)=OMSP(IQ,1,1)+(I-1)*DSE*CE(IQ)
          PB(NN,1,IQ)=PA(NN,1,IQ)
          PA(NN,2,IQ)=OMSP(IQ,4,1)+(I-1)*DSA*CA(IQ)
          PB(NN,2,IQ)=OMSP(IQ,4,2)+(I-1)*DSB*CB(IQ)
          PA(NN,3,IQ)=OMSP(IQ,4,1)+I*DSA*CA(IQ)
          PB(NN,3,IQ)=OMSP(IQ,4,2)+I*DSB*CB(IQ)
          PA(NN,4,IQ)=OMSP(IQ,1,1)+I*DSE*CE(IQ)
335       PB(NN,4,IQ)=PA(NN,4,IQ)
          IQUAD(NN)=0
          IF(CE(1)*CA(1)+CE(2)*CA(2)+CE(3)*CA(3).LT.0.999)GOTO 340
          IF(ABS(TA1).LT.0.03 .AND. ABS(TA2).LT.0.03)IQUAD(NN)=-1
340       IF(CE(1)*CB(1)+CE(2)*CB(2)+CE(3)*CB(3).LT.0.999)GOTO 330
          IF(ABS(TB1).LT.0.03 .AND. ABS(TB2).LT.0.03)IQUAD(NN)=
     2IQUAD(NN)-2
330       CONTINUE
          NOVT=NOVT+NOV
          ITK(IV)=NOV
205       CONTINUE
          RETURN
          END
```

APPENDIX 9

SUBROUTINE FGPOV

```
        SUBROUTINE FGPOV(NPA,ISA,NPB,ISB,OE,DOV,NCNRS,PCN,ICN,IPL,SEGM
     &,NDNPLT,PA,PB,IPLM,WV,TOUCH,TPSI,SPSI
     &,OMSP,NOV)
        DIMENSION OE(3,2),NCNRS(1),PCN(3,ICN,IPL),SEGM(1)
     &,NDNPLT(1),PA(IPLM,4,3),PB(IPLM,4,3),OMSP(3,4,2),OA(3,2),OB(3,2)
        DIMENSION CE(3),CA(3),CB(3),CA1(3),CA2(3),CB1(3),CB2(3),CP(3)
     &,CN(3),CMD(3),CMT(3)
        ICB=ISB
        ICB1=ISB+1
        IF(ISB.EQ.NCNRS(NPB))ICB1=1
        ICA=ISA
        ICA1=ISA+1
        IF(ISA.EQ.NCNRS(NPA))ICA1=1
        DAS=(PCN(1,ICA1,NPA)-PCN(1,ICA,NPA))**2
     2+(PCN(2,ICA1,NPA)-PCN(2,ICA,NPA))**2
     3+(PCN(3,ICA1,NPA)-PCN(3,ICA,NPA))**2
        DBS=(PCN(1,ICB1,NPB)-PCN(1,ICB,NPB))**2
     2+(PCN(2,ICB1,NPB)-PCN(2,ICB,NPB))**2
     3+(PCN(3,ICB1,NPB)-PCN(3,ICB,NPB))**2
C   SET UP MAXIMUM SEGMENT SIZES
        SEGA=SEGM(NPA)*WV
        SEGB=SEGM(NPB)*WV
210     NOVE=DOV/AMIN1(SEGA,SEGB)+0.99
C   CALCULATE DIRECTIONAL COSINES OF COMMON SIDE
        DO 241 IQ=1,3
241     CE(IQ)=(OE(IQ,2)-OE(IQ,1))/DOV
C   INDEX CORNERS OF PLATES
        IF(CE(1)*(PCN(1,ICA1,NPA)-PCN(1,ICA,NPA))
     2+CE(2)*(PCN(2,ICA1,NPA)-PCN(2,ICA,NPA))
     3+CE(3)*(PCN(3,ICA1,NPA)-PCN(3,ICA,NPA)).LT.0.)
     4GOTO 231
        ICA1=ISA
        ICA2=ICA1+1
        IF(ICA1.EQ.NCNRS(NPA))ICA2=1
        ICA3=ICA2+1
        IF(ICA2.EQ.NCNRS(NPA))ICA3=1
        ICA0=ICA1-1
        IF(ICA1.EQ.1)ICA0=NCNRS(NPA)
        GOTO 232
231     ICA2=ISA
```

```
        ICA1=ICA2+1
        IF(ICA2.EQ.NCNRS(NPA))ICA1=1
        ICA0=ICA1+1
        IF(ICA1.EQ.NCNRS(NPA))ICA0=1
        ICA3=ICA2-1
        IF(ICA2.EQ.1)ICA3=NCNRS(NPA)
232     IF(CE(1)*(PCN(1,ICB1,NPB)-PCN(1,ICB,NPB))
     2+CE(2)*(PCN(2,ICB1,NPB)-PCN(2,ICB,NPB))
     3+CE(3)*(PCN(3,ICB1,NPB)-PCN(3,ICB,NPB)).LT.0.)
     4GOTO 236
        ICB1=ISB
        ICB2=ICB1+1
        IF(ICB1.EQ.NCNRS(NPB))ICB2=1
        ICB3=ICB2+1
        IF(ICB2.EQ.NCNRS(NPB))ICB3=1
        ICB0=ICB1-1
        IF(ICB1.EQ.1)ICB0=NCNRS(NPB)
        GOTO 237
236     ICB2=ISB
        ICB1=ICB2+1
        IF(ICB2.EQ.NCNRS(NPB))ICB1=1
        ICB0=ICB1+1
        IF(ICB1.EQ.NCNRS(NPB))ICB0=1
        ICB3=ICB2-1
        IF(ICB2.EQ.1)ICB3=NCNRS(NPB)
C   FIND PRELIMINARY OVERLAP MODE REGIONS
237     DO 400 IQ=1,3
        OMSP(IQ,1,1)=PCN(IQ,ICA1,NPA)
        OMSP(IQ,2,1)=PCN(IQ,ICA2,NPA)
        OMSP(IQ,1,2)=PCN(IQ,ICB1,NPB)
400     OMSP(IQ,2,2)=PCN(IQ,ICB2,NPB)
        CALL FMDC(NDNPLT,PCN,ICN,IPL,PA,PB,IPLM,ICA2,ICA3,3,NPA
     &,1,CE,TOUCH,OMSP,WV,NPB)
        CALL FMDC(NDNPLT,PCN,ICN,IPL,PA,PB,IPLM,ICA1,ICA0,4,NPA
     &,1,CE,TOUCH,OMSP,WV,NPB)
        CALL FMDC(NDNPLT,PCN,ICN,IPL,PA,PB,IPLM,ICB2,ICB3,3,NPB
     &,2,CE,TOUCH,OMSP,WV,NPA)
        CALL FMDC(NDNPLT,PCN,ICN,IPL,PA,PB,IPLM,ICB1,ICB0,4,NPB
     &,2,CE,TOUCH,OMSP,WV,NPA)
        NCA1=1
        NCA2=1
        NCB1=1
        NCB2=1
        DOVAS=(OMSP(1,1,1)-OMSP(1,2,1))**2+(OMSP(2,1,1)-OMSP(2,2,1))**2
     &+(OMSP(3,1,1)-OMSP(3,2,1))**2
        DOVBS=(OMSP(1,1,1)-OMSP(1,2,2))**2+(OMSP(2,1,1)-OMSP(2,2,2))**2
     &+(OMSP(3,1,1)-OMSP(3,2,2))**2
        IF(DOVAS+DOVBS-2.*SQRT(DOVAS*DOVBS).LE.TOUCH**2)GOTO 500
        IF(DOVAS.LT.DOVBS)NCB2=0
        IF(DOVAS.GT.DOVBS)NCA2=0
        DO 500 IQ=1,3
        IF(DOVAS.LT.DOVBS)OMSP(IQ,2,2)=OMSP(IQ,2,1)
        IF(DOVAS.GT.DOVBS)OMSP(IQ,2,1)=OMSP(IQ,2,2)
500     CONTINUE
        DOVAS=(OMSP(1,2,1)-OMSP(1,1,1))**2+(OMSP(2,2,1)-OMSP(2,1,1))**2
     &+(OMSP(3,2,1)-OMSP(3,1,1))**2
        DOVBS=(OMSP(1,2,1)-OMSP(1,1,2))**2+(OMSP(2,2,1)-OMSP(2,1,2))**2
     &+(OMSP(3,2,1)-OMSP(3,1,2))**2
```

```
      IF(DOVAS+DOVBS-2.*SQRT(DOVAS*DOVBS).LE.TOUCH**2)GOTO 510
      IF(DOVAS.LT.DOVBS)NCB1=0
      IF(DOVAS.GT.DOVBS)NCA1=0
      DO 510 IQ=1,3
      IF(DOVAS.LT.DOVBS)OMSP(IQ,1,2)=OMSP(IQ,1,1)
      IF(DOVAS.GT.DOVBS)OMSP(IQ,1,1)=OMSP(IQ,1,2)
510   CONTINUE
      DOV=SQRT(AMIN1(DOVAS,DOVBS))
      DO 410 IQ=1,3
      CA(IQ)=OMSP(IQ,3,1)-OMSP(IQ,4,1)
410   CB(IQ)=OMSP(IQ,3,2)-OMSP(IQ,4,2)
      DA=SQRT(CA(1)*CA(1)+CA(2)*CA(2)+CA(3)*CA(3))
      DB=SQRT(CB(1)*CB(1)+CB(2)*CB(2)+CB(3)*CB(3))
      DO 420 IQ=1,3
      CA(IQ)=CA(IQ)/DA
420   CB(IQ)=CB(IQ)/DB
C  CALCULATE DIRECTIONAL COSINES OF ADJACENT SIDES
      DO 242 IQ=1,3
      CA1(IQ)=PCN(IQ,ICA0,NPA)-PCN(IQ,ICA1,NPA)
      CA2(IQ)=PCN(IQ,ICA3,NPA)-PCN(IQ,ICA2,NPA)
      CB1(IQ)=PCN(IQ,ICB0,NPB)-PCN(IQ,ICB1,NPB)
242   CB2(IQ)=PCN(IQ,ICB3,NPB)-PCN(IQ,ICB2,NPB)
      DA1=SQRT(CA1(1)**2+CA1(2)**2+CA1(3)**2)
      DA2=SQRT(CA2(1)**2+CA2(2)**2+CA2(3)**2)
      DB1=SQRT(CB1(1)**2+CB1(2)**2+CB1(3)**2)
      DB2=SQRT(CB2(1)**2+CB2(2)**2+CB2(3)**2)
      DO 243 IQ=1,3
      CA1(IQ)=CA1(IQ)/DA1
      CA2(IQ)=CA2(IQ)/DA2
      CB1(IQ)=CB1(IQ)/DB1
243   CB2(IQ)=CB2(IQ)/DB2
C  FIND PLATE CORNER ANGLES
      TA1=CE(1)*CA1(1)+CE(2)*CA1(2)+CE(3)*CA1(3)
      TA2=CE(1)*CA2(1)+CE(2)*CA2(2)+CE(3)*CA2(3)
      TB1=-(CE(1)*CB1(1)+CE(2)*CB1(2)+CE(3)*CB1(3))
      TB2=-(CE(1)*CB2(1)+CE(2)*CB2(2)+CE(3)*CB2(3))
      SA1=SQRT(1.-TA1*TA1)
      SA2=SQRT(1.-TA2*TA2)
      SB1=SQRT(1.-TB1*TB1)
      SB2=SQRT(1.-TB2*TB2)
C  DETERMINE FINAL OVERLAP MODE REGIONS
C  CALCULATE DIRECTIONAL COSINES OF NORMAL VECTOR, PLATE NPA
      CP(1)=(CE(2)*CA1(3)-CA1(2)*CE(3))/SA1
      CP(2)=(CE(3)*CA1(1)-CA1(3)*CE(1))/SA1
      CP(3)=(CE(1)*CA1(2)-CA1(1)*CE(2))/SA1
      CN(1)=CP(2)*CE(3)-CE(2)*CP(3)
      CN(2)=CP(3)*CE(1)-CE(3)*CP(1)
      CN(3)=CP(1)*CE(2)-CE(1)*CP(2)
C  FIND CORNER 1, PLATE NPA
      DO 425 IQ=1,3
      CMD(IQ)=CN(IQ)*SPSI-CE(IQ)*TPSI
425   CMT(IQ)=OMSP(IQ,4,1)-OMSP(IQ,1,1)
      DMT=SQRT(CMT(1)**2+CMT(2)**2+CMT(3)**2)
      TAL=(CMT(1)*CA(1)+CMT(2)*CA(2)+CMT(3)*CA(3))/DMT
      SAL=SQRT(1.-TAL*TAL)
      IF(NCA1.EQ.1)GOTO 245
      TBE=(CMT(1)*CMD(1)+CMT(2)*CMD(2)+CMT(3)*CMD(3))/DMT
      SBE=SQRT(1.-TBE*TBE)
      TBEP=(CMT(1)*CE(1)+CMT(2)*CE(2)+CMT(3)*CE(3))/DMT
      IF(TBEP.GE.-TPSI)GOTO 245
      DML=DMT*SAL/(SAL*TBE-TAL*SBE)
      IF(DML.GT.SEGA)GOTO 426
      DO 427 IQ=1,3
427   OA(IQ,1)=OMSP(IQ,1,1)+DML*CMD(IQ)
      DD1=(OA(1,1)-OMSP(1,4,1))**2+(OA(2,1)-OMSP(2,4,1))**2
     2+(OA(3,1)-OMSP(3,4,1))**2
      DD2=(OA(1,1)-OMSP(1,3,1))**2+(OA(2,1)-OMSP(2,3,1))**2
     2+(OA(3,1)-OMSP(3,3,1))**2
      IF(DD1+DD2-DA*DA.GT.2.*TOUCH**2)GOTO 245
      GOTO 250
426   DML=-DMT*TAL
      DML2=SEGA**2-(DMT*SAL)**2
      IF(DML2.GT.0.)DML=DML-SQRT(DML2)
      DO 428 IQ=1,3
428   OA(IQ,1)=OMSP(IQ,4,1)+DML*CA(IQ)
      GOTO 250
245   IF(DMT.GT.SEGA)GOTO 426
      DO 430 IQ=1,3
430   OA(IQ,1)=OMSP(IQ,4,1)
C   FIND CORNER 2, PLATE NPA
250   DO 435 IQ=1,3
      CMD(IQ)=CN(IQ)*SPSI+CE(IQ)*TPSI
435   CMT(IQ)=OMSP(IQ,3,1)-OMSP(IQ,2,1)
      DMT=SQRT(CMT(1)**2+CMT(2)**2+CMT(3)**2)
      TAL=(CMT(1)*CA(1)+CMT(2)*CA(2)+CMT(3)*CA(3))/DMT
      SAL=SQRT(1.-TAL*TAL)
      IF(NCA2.EQ.1)GOTO 255
      TBE=(CMT(1)*CMD(1)+CMT(2)*CMD(2)+CMT(3)*CMD(3))/DMT
      SBE=SQRT(1.-TBE*TBE)
      TBEP=(CMT(1)*CE(1)+CMT(2)*CE(2)+CMT(3)*CE(3))/DMT
      IF(TBEP.LT.TPSI)GOTO 255
      DML=DMT*SAL/(SAL*TBE+TAL*SBE)
      IF(DML.GT.SEGA)GOTO 436
      DO 437 IQ=1,3
437   OA(IQ,2)=OMSP(IQ,2,1)+DML*CMD(IQ)
      DD1=(OA(1,2)-OMSP(1,4,1))**2+(OA(2,2)-OMSP(2,4,1))**2
     2+(OA(3,2)-OMSP(3,4,1))**2
      DD2=(OA(1,2)-OMSP(1,3,2))**2+(OA(2,2)-OMSP(2,3,1))**2
     2+(OA(3,2)-OMSP(3,3,1))**2
      IF(DD1+DD2-DA*DA.GT.2.*TOUCH**2)GOTO 255
      GOTO 260
436   DML=DMT*TAL
      DML2=SEGA**2-(DMT*SAL)**2
      IF(DML2.GT.0.)DML=DML-SQRT(DML2)
      DO 438 IQ=1,3
438   OA(IQ,2)=OMSP(IQ,3,1)-DML*CA(IQ)
      GOTO 260
255   IF(DMT.GT.SEGA)GOTO 436
      DO 440 IQ=1,3
440   OA(IQ,2)=OMSP(IQ,3,1)
C   CALCULATE DIRECTIONAL COSINES OF NORMAL VECTOR, PLATE NPB
260   CP(1)=(CE(2)*CB1(3)-CB1(2)*CE(3))/SB1
      CP(2)=(CE(3)*CB1(1)-CB1(3)*CE(1))/SB1
      CP(3)=(CE(1)*CB1(2)-CB1(1)*CE(2))/SB1
      CN(1)=CP(2)*CE(3)-CE(2)*CP(3)
      CN(2)=CP(3)*CE(1)-CE(3)*CP(1)
```

```
        CN(3)=CP(1)*CE(2)-CE(1)*CP(2)
C   FIND CORNER 1, PLATE NPB
        DO 445 IQ=1,3
        CMD(IQ)=CN(IQ)*SPSI-CE(IQ)*TPSI
445     CMT(IQ)=OMSP(IQ,4,2)-OMSP(IQ,1,2)
        DMT=SQRT(CMT(1)**2+CMT(2)**2+CMT(3)**2)
        TAL=(CMT(1)*CB(1)+CMT(2)*CB(2)+CMT(3)*CB(3))/DMT
        SAL=SQRT(1.-TAL*TAL)
        IF(NCB1.EQ.1)GOTO 265
        TBE=(CMT(1)*CMD(1)+CMT(2)*CMD(2)+CMT(3)*CMD(3))/DMT
        SBE=SQRT(1.-TBE*TBE)
        TBEP=(CMT(1)*CE(1)+CMT(2)*CE(2)+CMT(3)*CE(3))/DMT
        IF(TBEP.GE.-TPSI)GOTO 265
        DML=DMT*SAL/(SAL*TBE-TAL*SBE)
        IF(DML.GT.SEGB)GOTO 446
        DO 447 IQ=1,3
447     OB(IQ,1)=OMSP(IQ,1,2)+DML*CMD(IQ)
        DD1=(OB(1,1)-OMSP(1,4,2))**2+(OB(2,1)-OMSP(2,4,2))**2
     2+(OB(3,1)-OMSP(3,4,2))**2
        DD2=(OB(1,1)-OMSP(1,3,2))**2+(OB(2,1)-OMSP(2,3,2))**2
     2+(OB(3,1)-OMSP(3,3,2))**2
        IF(DD1+DD2-DB*DB.GT.2.*TOUCH**2)GOTO 265
        GOTO 270
446     DML=-DMT*TAL
        DML2=SEGB**2-(DMT*SAL)**2
        IF(DML2.GT.0.)DML=DML-SQRT(DML2)
        DO 448 IQ=1,3
448     OB(IQ,1)=OMSP(IQ,4,2)+DML*CB(IQ)
        GOTO 270
265     IF(DMT.GT.SEGB)GOTO 446
        DO 450 IQ=1,3
450     OB(IQ,1)=OMSP(IQ,4,2)
C    FIND CORNER 2, PLATE NPB
270     DO 455 IQ=1,3
        CMD(IQ)=CN(IQ)*SPSI+CE(IQ)*TPSI
455     CMT(IQ)=OMSP(IQ,3,2)-OMSP(IQ,2,2)
        DMT=SQRT(CMT(1)**2+CMT(2)**2+CMT(3)**2)
        TAL=(CMT(1)*CB(1)+CMT(2)*CB(2)+CMT(3)*CB(3))/DMT
        SAL=SQRT(1.-TAL*TAL)
        IF(NCB2.EQ.1)GOTO 275
        TBE=(CMT(1)*CMD(1)+CMT(2)*CMD(2)+CMT(3)*CMD(3))/DMT
        SBE=SQRT(1.-TBE*TBE)
        TBEP=(CMT(1)*CE(1)+CMT(2)*CE(2)+CMT(3)*CE(3))/DMT
        IF(TBEP.LT.TPSI)GOTO 275
        DML=DMT*SAL/(SAL*TBE+TAL*SBE)
        IF(DML.GT.SEGB)GOTO 456
        DO 457 IQ=1,3
457     OB(IQ,2)=OMSP(IQ,2,2)+DML*CMD(IQ)
        DD1=(OB(1,2)-OMSP(1,4,2))**2+(OB(2,2)-OMSP(2,4,2))**2
     2+(OB(3,2)-OMSP(3,4,2))**2
        DD2=(OB(1,2)-OMSP(1,3,2))**2+(OB(2,2)-OMSP(2,3,2))**2
     2+(OB(3,2)-OMSP(3,3,2))**2
        IF(DD1+DD2-DB*DB.GT.2.*TOUCH**2)GOTO 275
        GOTO 280
456     DML=DMT*TAL
        DML2=SEGB**2-(DMT*SAL)**2
        IF(DML2.GT.0.)DML=DML-SQRT(DML2)
        DO 458 IQ=1,3
```

```
458     OB(IQ,2)=OMSP(IQ,3,2)-DML*CB(IQ)
        GOTO 280
275     IF(DMT.GT.SEGB)GOTO 456
        DO 460 IQ=1,3
460     OB(IQ,2)=OMSP(IQ,3,2)
C   DETERMINE NUMBER OF OVERLAP MODES
280     DA=SQRT((OA(1,2)-OA(1,1))**2+(OA(2,2)-OA(2,1))**2
     2+(OA(3,2)-OA(3,1))**2)
        NOVA=DA/SEGA+.99
        DB=SQRT((OB(1,2)-OB(1,1))**2+(OB(2,2)-OB(2,1))**2
     2+(OB(3,2)-OB(3,1))**2)
        NOVB=DB/SEGB+.99
        NOV=JMAX0(NOVE,NOVA,NOVB)
        DO 515 IQ=1,3
        OMSP(IQ,3,1)=OA(IQ,2)
        OMSP(IQ,4,1)=OA(IQ,1)
        OMSP(IQ,3,2)=OB(IQ,2)
515     OMSP(IQ,4,2)=OB(IQ,1)
        RETURN
        END
```

APPENDIX 10

SUBROUTINE FMDC

```
        SUBROUTINE FMDC(NDNPLT,PCN,ICN,IPL,PA,PB,IPLM,NC,NAC,MC,NP
     &,IAB,CE,TOUCH,OMSP,WV,NPO)
C      SUBROUTINE FMDC FINDS THE OVERLAP REGION POINT OMSP(I,MC,IAB)
        DIMENSION NDNPLT(1),PCN(3,ICN,IPL),PA(IPLM,4,3),PB(IPLM,4,3)
     &,CE(3),OMSP(3,4,2),VM(3)
        TCHS=TOUCH**2
        DMN=WV*WV
        MAB=0
        M0=1
        IF(NP.NE.1)M0=NDNPLT(NP-1)+1
        M1=NDNPLT(NP)
        IF(M0-1.EQ.M1)GOTO 100
        DO 110 M=M0,M1
C      CHECK MONOPOLE A OF MODE M
        DO 120 IQ=1,3
120     VM(IQ)=PA(M,3,IQ)-PA(M,2,IQ)
        DES=VM(1)*VM(1)+VM(2)*VM(2)+VM(3)*VM(3)-(VM(1)*CE(1)
     2+VM(2)*CE(2)+VM(3)*CE(3))**2
        IF(DES.GT.TCHS)GOTO 130
C      CHECK MONOPOLE A, CORNER 2
        DCS=(PCN(1,NC,NP)-PA(M,2,1))**2+(PCN(2,NC,NP)-PA(M,2,2))**2
     2+(PCN(3,NC,NP)-PA(M,2,3))**2
        IF(DCS.LE.TCHS)GOTO 145
        IF(DMN.LT.DCS)GOTO 140
        DMN=DCS
        MDN=M
        MAB=1
        MCO=2
        GOTO 140
145     DO 150 IQ=1,3
150     OMSP(IQ,MC,IAB)=PA(M,1,IQ)
        RETURN
C      CHECK MONOPOLE A, CORNER 3
140     DCS=(PCN(1,NC,NP)-PA(M,3,1))**2+(PCN(2,NC,NP)-PA(M,3,2))**2
     2+(PCN(3,NC,NP)-PA(M,3,3))**2
        IF(DCS.LE.TCHS)GOTO 155
        IF(DMN.LT.DCS)GOTO 130
        DMN=DCS
        MDN=M
        MAB=1
```

```
      MCO=3
      GOTO 130
155   DO 160 IQ=1,3
160   OMSP(IQ,MC,IAB)=PA(M,4,IQ)
      RETURN
C    CHECK MONOPOLE B OF MODE M
130   DO 170 IQ=1,3
170   VM(IQ)=PB(M,3,IQ)-PB(M,2,IQ)
      DES=VM(1)*VM(1)+VM(2)*VM(2)+VM(3)*VM(3)-(VM(1)*CE(1)
     2+VM(2)*CE(2)+VM(3)*CE(3))**2
      IF(DES.GT.TCHS)GOTO 110
C    CHECK MONOPOLE B, CORNER 2
      DCS=(PCN(1,NC,NP)-PB(M,2,1))**2+(PCN(2,NC,NP)-PB(M,2,2))**2
     2+(PCN(3,NC,NP)-PB(M,2,3))**2
      IF(DCS.LE.TCHS)GOTO 185
      IF(DMN.LT.DCS)GOTO 180
      DMN=DCS
      MDN=M
      MAB=2
      MCO=2
      GOTO 180
185   DO 190 IQ=1,3
190   OMSP(IQ,MC,IAB)=PB(M,1,IQ)
      RETURN
C    CHECK MONOPOLE B, CORNER 3
180   DCS=(PCN(1,NC,NP)-PB(M,3,1))**2+(PCN(2,NC,NP)-PB(M,3,2))**2
     2+(PCN(3,NC,NP)-PB(M,3,3))**2
      IF(DCS.LE.TCHS)GOTO 195
      IF(DMN.LT.DCS)GOTO 110
      DMN=DCS
      MDN=M
      MAB=2
      MCO=3
      GOTO 110
195   DO 200 IQ=1,3
200   OMSP(IQ,MC,IAB)=PB(M,4,IQ)
      RETURN
110   CONTINUE
      IF(MAB.EQ.0)GOTO 100
C    CORNER MCO OF MONOPOLE MAB OF MODE MDN IS CLOSEST TO CORNER NC
C     OF PLATE NP
      IF(DMN/WV/WV.GT.0.125)WRITE(6,1)NP,NPO
1     FORMAT(//'  **********  POSSIBLE PROBLEM WITH OVERLAP',
     2' MODES BETWEEN PLATES ',I3,' AND ',I3)
      MCO2=1
      IF(MCO.EQ.3)MCO2=4
      MC2=1
      IF(MC.EQ.3)MC2=2
      DO 310 IQ=1,3
      IF(MAB.EQ.2)GOTO 300
      OMSP(IQ,MC2,IAB)=PA(MDN,MCO,IQ)
      OMSP(IQ,MC,IAB)=PA(MDN,MCO2,IQ)
      GOTO 310
300   OMSP(IQ,MC2,IAB)=PB(MDN,MCO,IQ)
      OMSP(IQ,MC,IAB)=PB(MDN,MCO2,IQ)
310   CONTINUE
      RETURN
100   DO 210 IQ=1,3
210   OMSP(IQ,MC,IAB)=PCN(IQ,NAC,NP)
      RETURN
      END
```

APPENDIX 11

SUBROUTINE MPLOT

```
        SUBROUTINE MPLOT(NCNRS,PCN,NPL,ICN,IPL,IPLM,NPL11,
      1 NPL22,NDNPLT,PA,PB,IPN)
        DIMENSION PCN(3,ICN,IPL),X(20),Y(20),Z(20),PHI(20),IPN(1),
      1 SXY(20),RZ(20),SXZ(20),APHIY(20),APHI(20),PHIY(20)
      2 ,NCNRS(1),NPL11(1),NPL22(1),NDNPLT(1),PA(IPLM,4,3),
      3 PB(IPLM,4,3),XS(10),YS(10),ZS(10),TNPL(20)
        SIZE=2.5
        NP=NPL
        XNPL=NPL
        NC=NCNRS(NPL)
        PI=3.141592
        IF(NPL.EQ.1)TOT=NDNPLT(NPL)
        IF(NPL.NE.1)TOT=NDNPLT(NPL)-NDNPLT(NPL-1)
        DO 1 I=1,NC
        X(I)=PCN(1,I,NPL)-PCN(1,1,NPL)
        Y(I)=PCN(2,I,NPL)-PCN(2,1,NPL)
        Z(I)=PCN(3,I,NPL)-PCN(3,1,NPL)
1       CONTINUE
        CX=-Y(2)*Z(3)+Z(2)*Y(3)
        CY=X(2)*Z(3)-X(3)*Z(2)
        CZ=-X(2)*Y(3)+X(3)*Y(2)
        CMAG=(CX**2+CY**2+CZ**2)**0.5
        THC=ACOS(CZ/CMAG)
        IF(CX.EQ.0.0.AND.CY.EQ.0.0)GO TO 800
        IF(CX.EQ.0.0.AND.CY.NE.0.0)PHC=(CY/SQRT(CY**2))
      1 *(PI/2)
        IF(CX.NE.0.0)PHC=ATAN2(CY,CX)
        DO 10 I=2,NC
        SXY(I)=(X(I)**2+Y(I)**2)**0.5
        IF(X(I).EQ.0.0.AND.Y(I).EQ.0.0)GO TO 13
        IF(X(I).EQ.0.0.AND.Y(I).NE.0.0)GO TO 11
        GO TO 12
11      M=Y(I)/((Y(I)**2)**0.5)
        PHI(I)=(PI/2)*M-PHC
        GO TO 10
12      PHI(I)=ATAN2(Y(I),X(I))
        PHI(I)=PHI(I)-PHC
        GO TO 10
13      CONTINUE
10      CONTINUE
```

```
      DO 20 I=2,NC
      IF(X(I).EQ.0.0.AND.Y(I).EQ.0.0)GO TO 14
      GO TO 15
14    X(I)=0.0
      Y(I)=0.0
15    X(I)=SXY(I)*COS(PHI(I))
      Y(I)=SXY(I)*SIN(PHI(I))
      SXZ(I)=(X(I)**2+Z(I)**2)**0.5
      IF(Z(I).EQ.0.0.AND.ABS(X(I)).LT.0.0005)GO TO 30
      GO TO 31
30    X(I)=0.0
      Z(I)=0.0
      GO TO 20
31    PHIY(I)=ATAN2(X(I),Z(I))
      PHIY(I)=PHIY(I)-THC
      X(I)=SXZ(I)*SIN(PHIY(I))
      Y(I)=Y(I)
      Z(I)=SXZ(I)*COS(PHIY(I))
20    CONTINUE
800   X1=0.0
      Y1=0.0
      Z1=0.0
      DO 2 I=1,NC
      X1=X1+X(I)
      Y1=Y1+Y(I)
      Z1=Z1+Z(I)
2     CONTINUE
      X1=X1/NC
      Y1=Y1/NC
      Z1=Z1/NC
C     CALL VPLOTS(0,0,0)
      SM=-1.0
      DO 110 I=1,NC
      DO 120 J=1,NC
      IF(I.EQ.J) GO TO 120
      SL=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
      IF(SL.GT.SM)SM=SL
120   CONTINUE
110   CONTINUE
      FAC=SIZE/SM
      DO 130 I=1,NC
      X(I)=(X(I)-X1)*FAC
      Y(I)=(Y(I)-Y1)*FAC
      Z(I)=(Z(I)-Z1)*FAC
130   CONTINUE
      CMAX=-999.0
      DO 16 I=1,NC
      IF(X(I).GT.CMAX)CMAX=X(I)
16    CONTINUE
      CALL PLOT(3.0,0.75,-3)
      CALL SYMBOL(-1.5,-0.5,0.2,20HTOTAL MODES ON PLATE,0.0,20)
      CALL NUMBER(-2.30,-0.5,0.2,TOT,0.0,-1)
      CALL NUMBER(2.9,-0.5,0.2.XNPL,0.0,-1)
      NPS=NC+1
      X(NPS)=X(1)
      Y(NPS)=Y(1)
      DO 1100 K=1,2
      YY=3.0
```

```
        XX1=2.0
        XX2=1.5
        IF(K.EQ.1)YY=1.5
        YY1=0.3
        CALL PLOT(0.0,YY,-3)
        CALL SYMBOL(XX1,YY1,0.2,5HMODES,0.0,5)
        DO 100 N=1,NPS
        IU=3
        IF(N.GT.1)IU=2
1999    FORMAT(1X,' X(N) = ',F6.2,2X,' Y(N) = ',F6.2)
100     CALL PLOT(X(N),Y(N),IU)
        IF(NPL.EQ.1)GO TO 444
        GO TO 445
444     IF(K.EQ.1)I0=1
        IF(K.EQ.1)I2=NPL11(NPL)
        IF(K.EQ.1)AM=NPL11(NPL)
        IF(K.EQ.2)AM=NPL22(NPL)-NPL11(NPL)
        IF(K.EQ.1.OR.K.EQ.2)CALL NUMBER(XX2,YY1,0.2,AM,0.0,-1)
        IF(K.EQ.2.AND.NDNPLT(NPL).EQ.NPL11(NPL))GO TO 1100
        IF(K.EQ.2)I0=NPL11(NPL)+1
        IF(K.EQ.2)I2=NPL22(NPL)
        IF(K.EQ.3)AM=NDNPLT(NPL)-NPL22(NPL)
        IF(K.EQ.3)CALL NUMBER(XX2,YY1,0.2,AM,0.0,-1)
        IF(K.EQ.3.AND.NDNPLT(NPL).EQ.NPL11(NPL))GO TO 1100
        IF(K.EQ.3.AND.NDNPLT(NPL).EQ.NPL22(NPL))GO TO 1100
        IF(K.EQ.3)I0=NPL22(NPL)+1
        IF(K.EQ.3)I2=NDNPLT(NPL)
        GO TO 446
445     IF(K.EQ.1)I0=NDNPLT(NPL-1)+1
        IF(K.EQ.1)I2=NDNPLT(NPL-1)+NPL11(NPL)
        IF(K.EQ.1)AM=NPL11(NPL)
        IF(K.EQ.2)AM=NPL22(NPL)-NPL11(NPL)
        IF(IPN(NPL).EQ.0)AM=0
        IF(K.EQ.1.OR.K.EQ.2)CALL NUMBER(XX2,YY1,0.2,AM,0.0,-1)
        IF(K.EQ.2.AND.NDNPLT(NPL).EQ.(NDNPLT(NPL-1)+NPL11(NPL)))GO TO 1100
        IF(K.EQ.2)I0=NDNPLT(NPL-1)+NPL11(NPL)+1
        IF(K.EQ.2)I2=NDNPLT(NPL-1)+NPL22(NPL)
        IF(K.EQ.3)AM=(NDNPLT(NPL)-NDNPLT(NPL-1))-NPL22(NPL)
        IF(IPN(NPL).EQ.0)AM=0
        IF(K.EQ.3)CALL NUMBER(XX2,YY1,0.2,AM,0.0,-1)
        IF(K.EQ.3.AND.NDNPLT(NPL).EQ.(NDNPLT(NPL-1)+NPL11(NPL)))GO TO 1100
        IF(K.EQ.3.AND.NDNPLT(NPL).EQ.(NDNPLT(NPL-1)+NPL22(NPL)))GO TO 1100
        IF(K.EQ.3)I0=NDNPLT(NPL-1)+NPL22(NPL)+1
        IF(K.EQ.3)I2=NDNPLT(NPL)
446     IF(IPN(NPL).EQ.0)GO TO 1100
        DO 1000 I=I0,I2
        IF(I0.GT.I2) GO TO 1000
        DO 1002 IAB=1,2
        DO 1001 J=1,4
        IF(IAB.EQ.2)GO TO 501
        XS(J)=PA(I,J,1)-PCN(1,1,NPL)
        YS(J)=PA(I,J,2)-PCN(2,1,NPL)
        ZS(J)=PA(I,J,3)-PCN(3,1,NPL)
        GO TO 1001
501     XS(J)=PB(I,J,1)-PCN(1,1,NPL)
        YS(J)=PB(I,J,2)-PCN(2,1,NPL)
        ZS(J)=PB(I,J,3)-PCN(3,1,NPL)
1001    CONTINUE
```

```
          IF(CX.EQ.0.0.AND.CY.EQ.0.0)GO TO 900
600       DO 1003 M=1,4
          SXY(M)=SQRT(XS(M)**2+YS(M)**2)
          IF(XS(M).EQ.0.0.AND.YS(M).EQ.0.0)GO TO 35
          IF(XS(M).EQ.0.0.AND.YS(M).NE.0.0)GO TO 36
          PHI(M)=ATAN2(YS(M),XS(M))-PHC
          GO TO 400
36        M1=YS(M)/((YS(M)**2)**0.5)
          PHI(M)=(PI/2)*M1-PHC
          GO TO 400
35        XS(M)=XS(M)
          YS(M)=YS(M)
          ZS(M)=ZS(M)
          GO TO 1003
400       XS(M)=SXY(M)*COS(PHI(M))
          YS(M)=SXY(M)*SIN(PHI(M))
          ZS(M)=ZS(M)
1003      CONTINUE
          DO 1004 J1=1,4
          SXZ(J1)=SQRT(XS(J1)**2 + ZS(J1)**2)
          IF(ZS(J1).EQ.0.0.AND.ABS(XS(J1)).LT.0.0005)GO TO 37
          PHIY(J1)=ATAN2(XS(J1),ZS(J1))-THC
          GO TO 401
37        XS(J1)=0.0
          ZS(J1)=0.0
          YS(J1)=YS(J1)
          GO TO 1004
38        XS(J1)=XS(J1)
          ZS(J1)=0.0
          YS(J1)=YS(J1)
          GO TO 1004
401       XS(J1)=SXZ(J1)*SIN(PHIY(J1))
          ZS(J1)=SXZ(J1)*COS(PHIY(J1))
          YS(J1)=YS(J1)
1004      CONTINUE
900       DO 1111 JK=1,4
          XS(JK)=(XS(JK)-X1)*FAC
          YS(JK)=(YS(JK)-Y1)*FAC
1111      ZS(JK)=(ZS(JK)-Z1)*FAC
          KPS=5
          XS(KPS)=XS(1)
          YS(KPS)=YS(1)
          ZS(KPS)=ZS(1)
334       FORMAT(1X,'THE FINAL COOR. ARE:',//)
          DO 1005 I1=1,5
          IU=3
          IF(I1.GT.1)IU=2
1005      CALL PLOT(XS(I1),YS(I1),IU)
          XAB=0.0
          YAB=0.0
          DO 1006 L=1,4
          XAB=XAB+XS(L)
          YAB=YAB+YS(L)
1006      CONTINUE
          XAB=XAB/4
          YAB=YAB/4
          DX=(XS(1)+XS(4))/2
          DY=(YS(1)+YS(4))/2
```

```
      DXCD=DX-XAB
      DYCD=DY-YAB
      CPX=XAB+0.1*DXCD
      CPY=YAB+0.1*DYCD
      CALL PLOT(CPX,CPY,3)
      CALL PLCT(DX,DY,2)
      IF(IAB.EQ.2)GO TO 700
      GO TO 1002
700   DIFX=DX-CPX
      DIFY=DY-CPY
      DIF=SQRT(DIFX**2+DIFY**2)
      IF(ABS(DIFX).LT.0.005)THETA=(PI/2)*DIFY/ABS(DIFY)
      IF(ABS(DIFX).GE.0.005)THETA=ATAN2(DIFY,DIFX)
      CPE=0.30*DIF
222   FORMAT(1X,'THETA IS = ',F12.6)
      EX1=CPX+CPE*COS(THETA-PI/6)
      EY1=CPY+CPE*SIN(THETA-PI/6)
      EX2=CPX+CPE*COS(THETA+PI/6)
      EY2=CPY+CPE*SIN(THETA+PI/6)
      CALL PLOT(CPX,CPY,3)
      CALL PLOT(EX1,EY1,2)
      CALL PLOT(CPX,CPY,3)
      CALL PLOT(EX2,EY2,2)
1002  CONTINUE
1000  CONTINUE
1100  CONTINUE
      CALL PLOT(0.0,0.0,-999)
      RETURN
      END
```

APPENDIX 12

SUBROUTINE MOPLOT

```
      SUBROUTINE MOPLOT(PCN,NCNRS,IPL,ICN,PA,PB,IPLM
     2,IOVT,ITK,NOPL,NPLTM,NOVT)
      DIMENSION PCN(3,ICN,IPL),NCNRS(IPL),PA(IPLM,4,3)
     2,PB(IPLM,4,3),IOVT(IPLM,4),ITK(IPLM),OE(3,2)
      IF(NOPL.EQ.0)RETURN
      ICT=0
      SIZ=2.5
      DO 10 IV=1,NOPL
      IF(ITK(IV).EQ.0)GOTO 10
      CALL PLOT(4.0,5.25,-3)
      NPA=IOVT(IV,1)
      ISA=IOVT(IV,2)
      NPB=IOVT(IV,3)
      ISB=IOVT(IV,4)
      ICA1=ISA
      ICA2=ICA1+1
      IF(ICA1.EQ.NCNRS(NPA))ICA2=1
      ICA0=ICA1-1
      IF(ICA1.EQ.1)ICA0=NCNRS(NPA)
      ICB1=ISB
      ICB2=ICB1+1
      IF(ICB1.EQ.NCNRS(NPB))ICB2=1
      ICB0=ICB1-1
      IF(ICB1.EQ.1)ICB0=NCNRS(NPB)
      DAS=(PCN(1,ICA2,NPA)-PCN(1,ICA1,NPA))**2
     2+(PCN(2,ICA2,NPA)-PCN(2,ICA1,NPA))**2
     3+(PCN(3,ICA2,NPA)-PCN(3,ICA1,NPA))**2
      DBS=(PCN(1,ICB2,NPB)-PCN(1,ICB1,NPB))**2
     2+(PCN(2,ICB2,NPB)-PCN(2,ICB1,NPB))**2
     3+(PCN(3,ICB2,NPB)-PCN(3,ICB1,NPB))**2
      IOV=0
      DO 40 IC=1,2
      ICB=ICB1
      IF(IC.EQ.2)ICB=ICB2
      DCJL=(PCN(1,ICA1,NPA)-PCN(1,ICB,NPB))**2
     2+(PCN(2,ICA1,NPA)-PCN(2,ICB,NPB))**2
     3+(PCN(3,ICA1,NPA)-PCN(3,ICB,NPB))**2
      DCJH=(PCN(1,ICA2,NPA)-PCN(1,ICB,NPB))**2
     2+(PCN(2,ICA2,NPA)-PCN(2,ICB,NPB))**2
     3+(PCN(3,ICA2,NPA)-PCN(3,ICB,NPB))**2
```

```
      IF(DCJL+DCJH-DAS.GT.1.E-5)GOTO 40
      IOV=IOV+1
      DO 50 IQ=1,3
50    OE(IQ,IOV)=PCN(IQ,ICB,NPB)
40    CONTINUE
      IF(IOV.EQ.2)DOV=SQRT((OE(1,2)-OE(1,1))**2
     2+(OE(2,2)-OE(2,1))**2+(OE(3,2)-OE(3,1))**2)
      IF(IOV.EQ.2)GOTO 60
      DO 70 IC=1,2
      ICA=ICA1
      IF(IC.EQ.2)ICA=ICA2
      DCJL=(PCN(1,ICB1,NPB)-PCN(1,ICA,NPA))**2
     2+(PCN(2,ICB1,NPB)-PCN(2,ICA,NPA))**2
     3+(PCN(3,ICB1,NPB)-PCN(3,ICA,NPA))**2
      DCJH=(PCN(1,ICB2,NPB)-PCN(1,ICA,NPA))**2
     2+(PCN(2,ICB2,NPB)-PCN(2,ICA,NPA))**2
     3+(PCN(3,ICB2,NPB)-PCN(3,ICA,NPA))**2
      IF(DCJL+DCJH-DBS.GT.1.E-5)GOTO 70
      IOV=IOV+1
      DO 80 IQ=1,3
80    OE(IQ,IOV)=PCN(IQ,ICA,NPA)
      IF(IOV.NE.2)GOTO 70
      DOV=SQRT((OE(1,2)-OE(1,1))**2
     2+(OE(2,2)-OE(2,1))**2+(OE(3,2)-OE(3,1))**2)
      IF(DOV.GT.1.E-5)GOTO 60
      IOV=1
70    CONTINUE
60    TPX=(OE(1,1)+OE(1,2))/2.
      TPY=(OE(2,1)+OE(2,2))/2.
      TPZ=(OE(3,1)+OE(3,2))/2.
      SG=0.0
      DO 20 IC=1,NCNRS(NPA)
      SL=(PCN(1,IC,NPA)-TPX)**2+(PCN(2,IC,NPA)-TPY)**2
     2+(PCN(3,IC,NPA)-TPZ)**2
      IF(SL.GT.SG)SG=SL
20    CONTINUE
      DO 30 IC=1,NCNRS(NPB)
      SL=(PCN(1,IC,NPB)-TPX)**2+(PCN(2,IC,NPB)-TPY)**2
     2+(PCN(3,IC,NPB)-TPZ)**2
      IF(SL.GT.SG)SG=SL
30    CONTINUE
      FAC=SIZ/SQRT(SG)
      CXX=(OE(1,2)-OE(1,1))/DOV
      CYX=(OE(2,2)-OE(2,1))/DOV
      CZX=(OE(3,2)-OE(3,1))/DOV
      DA=SQRT(DAS)
      AXX=(PCN(1,ICA2,NPA)-PCN(1,ICA1,NPA))/DA
      AYX=(PCN(2,ICA2,NPA)-PCN(2,ICA1,NPA))/DA
      AZX=(PCN(3,ICA2,NPA)-PCN(3,ICA1,NPA))/DA
      IF(AXX*CXX+AYX*CYX+AZX*CZX.GT.0.)GOTO 90
      ICA1=ICA2
      ICA2=ISA
      ICA0=ICA1+1
      IF(ICA1.EQ.NCNRS(NPA))ICA0=1
90    DB=SQRT(DBS)
      BXX=(PCN(1,ICB2,NPB)-PCN(1,ICB1,NPB))/DB
      BYX=(PCN(2,ICB2,NPB)-PCN(2,ICB1,NPB))/DB
      BZX=(PCN(3,ICB2,NPB)-PCN(3,ICB1,NPB))/DB
      IF(BXX*CXX+BYX*CYX+BZX*CZX.GT.0.)GOTO 100
      ICB1=ICB2
      ICB2=ISB
      ICB0=ICB1+1
      IF(ICB1.EQ.NCNRS(NPB))ICB0=1
100   PAX=PCN(1,ICA1,NPA)-PCN(1,ICA0,NPA)
      PAY=PCN(2,ICA1,NPA)-PCN(2,ICA0,NPA)
      PAZ=PCN(3,ICA1,NPA)-PCN(3,ICA0,NPA)
      PAS=SQRT(PAX*PAX+PAY*PAY+PAZ*PAZ)
      PAX=PAX/PAS
      PAY=PAY/PAS
      PAZ=PAZ/PAS
      AXZ=PAY*CZX-PAZ*CYX
      AYZ=PAZ*CXX-PAX*CZX
      AZZ=PAX*CYX-PAY*CXX
      A=SQRT(AXZ*AXZ+AYZ*AYZ+AZZ*AZZ)
      AXZ=AXZ/A
      AYZ=AYZ/A
      AZZ=AZZ/A
      AXY=AYZ*CZX-AZZ*CYX
      AYY=AZZ*CXX-AXZ*CZX
      AZY=AXZ*CYX-AYZ*CXX
      PBX=PCN(1,ICB0,NPB)-PCN(1,ICB1,NPB)
      PBY=PCN(2,ICB0,NPB)-PCN(2,ICB1,NPB)
      PBZ=PCN(3,ICB0,NPB)-PCN(3,ICB1,NPB)
      PBS=SQRT(PBX*PBX+PBY*PBY+PBZ*PBZ)
      PBX=PBX/PBS
      PBY=PBY/PBS
      PBZ=PBZ/PBS
      BXZ=PBY*CZX-PBZ*CYX
      BYZ=PBZ*CXX-PBX*CZX
      BZZ=PBX*CYX-PBY*CXX
      B=SQRT(BXZ*BXZ+BYZ*BYZ+BZZ*BZZ)
      BXZ=BXZ/B
      BYZ=BYZ/B
      BZZ=BZZ/B
      BXY=BYZ*CZX-BZZ*CYX
      BYY=BZZ*CXX-BXZ*CZX
      BZY=BXZ*CYX-BYZ*CXX
      IU=3
      DO 110 IC=1,NCNRS(NPA)+1
      IF(IC.NE.1)IU=2
      JC=IC
      IF(IC.EQ.NCNRS(NPA)+1)JC=1
      X=(CXX*(PCN(1,JC,NPA)-TPX)+CYX*(PCN(2,JC,NPA)-TPY)
     2+CZX*(PCN(3,JC,NPA)-TPZ))*FAC
      Y=(AXY*(PCN(1,JC,NPA)-TPX)+AYY*(PCN(2,JC,NPA)-TPY)
     2+AZY*(PCN(3,JC,NPA)-TPZ))*FAC
110   CALL PLOT(X,Y,IU)
      IU=3
      DO 120 IC=1,NCNRS(NPB)+1
      IF(IC.NE.1)IU=2
      JC=IC
      IF(IC.EQ.NCNRS(NPB)+1)JC=1
      X=(CXX*(PCN(1,JC,NPB)-TPX)+CYX*(PCN(2,JC,NPB)-TPY)
     2+CZX*(PCN(3,JC,NPB)-TPZ))*FAC
      Y=(BXY*(PCN(1,JC,NPB)-TPX)+BYY*(PCN(2,JC,NPB)-TPY)
     2+BZY*(PCN(3,JC,NPB)-TPZ))*FAC
```

```
120        CALL PLOT(X,Y,IU)
           DO 130 IM=1,ITK(IV)
           II=NPLTM-NOVT+ICT+IM
           IU=3
           DO 140 IC=1,5
           JC=IC
           IF(IC.EQ.5)JC=1
           IF(IC.NE.1)IU=2
           X=(CXX*(PA(II,JC,1)-TPX)+CYX*(PA(II,JC,2)-TPY)
      2+CZX*(PA(II,JC,3)-TPZ))*FAC
           Y=(AXY*(PA(II,JC,1)-TPX)+AYY*(PA(II,JC,2)-TPY)
      2+AZY*(PA(II,JC,3)-TPZ))*FAC
140        CALL PLOT(X,Y,IU)
           IU=3
           DO 150 IC=1,5
           JC=IC
           IF(IC.EQ.5)JC=1
           IF(IC.NE.1)IU=2
           X=(CXX*(PB(II,JC,1)-TPX)+CYX*(PB(II,JC,2)-TPY)
      2+CZX*(PB(II,JC,3)-TPZ))*FAC
           Y=(BXY*(PB(II,JC,1)-TPX)+BYY*(PB(II,JC,2)-TPY)
      2+BZY*(PB(II,JC,3)-TPZ))*FAC
150        CALL PLOT(X,Y,IU)
           XMA=(PA(II,1,1)+PA(II,2,1)+PA(II,3,1)+PA(II,4,1))/4.
           YMA=(PA(II,1,2)+PA(II,2,2)+PA(II,3,2)+PA(II,4,2))/4.
           ZMA=(PA(II,1,3)+PA(II,2,3)+PA(II,3,3)+PA(II,4,3))/4.
           XMB=(PB(II,1,1)+PB(II,2,1)+PB(II,3,1)+PB(II,4,1))/4.
           YMB=(PB(II,1,2)+PB(II,2,2)+PB(II,3,2)+PB(II,4,2))/4.
           ZMB=(PB(II,1,3)+PB(II,2,3)+PB(II,3,3)+PB(II,4,3))/4.
           XME=(PB(II,1,1)+PB(II,4,1))/2.
           YME=(PB(II,1,2)+PB(II,4,2))/2.
           ZME=(PB(II,1,3)+PB(II,4,3))/2.
           X=(CXX*(XMA-TPX)+CYX*(YMA-TPY)+CZX*(ZMA-TPZ))*FAC
           Y=(AXY*(XMA-TPX)+AYY*(YMA-TPY)+AZY*(ZMA-TPZ))*FAC
           CALL PLOT(X,Y,3)
           XE=(CXX*(XME-TPX)+CYX*(YME-TPY)+CZX*(ZME-TPZ))*FAC
           CALL PLOT(XE,0.0,2)
           XB=(CXX*(XMB-TPX)+CYX*(YMB-TPY)+CZX*(ZMB-TPZ))*FAC
           YB=(BXY*(XMB-TPX)+BYY*(YMB-TPY)+BZY*(ZMB-TPZ))*FAC
           CALL PLOT(XB,YB,2)
           DP=SQRT((XB-XE)**2+YB*YB)
           CXP=(XB-XE)/DP
           CYP=YB/DP
           X=DP*(CXP*0.74+CYP*0.15)+XE
           Y=DP*(CYP*0.74-CXP*0.15)
           CALL PLOT(X,Y,2)
           CALL PLOT(XB,YB,3)
           X=DP*(CXP*0.74-CYP*0.15)+XE
           Y=DP*(CYP*0.74+CXP*0.15)
           CALL PLOT(X,Y,2)
130        CONTINUE
           ICT=ICT+ITK(IV)
           CALL NUMBER(-3.25,-3.6,.2,FLOAT(ITK(IV)),0.,-1)
           CALL SYMBOL(-2.65,-3.6,.2,'OVERLAP MODES BETWEEN',0.,21)
           CALL SYMBOL(-3.25,-3.95,.2,'PLATE   , SIDE    AND',0.,21)
           CALL NUMBER(-2.05,-3.95,.2,FLOAT(NPA),0.,-1)
           CALL NUMBER(-.25,-3.95,.2,FLOAT(ISA),0.,-1)
           CALL SYMBOL(-3.25,-4.3,.2,'PLATE   , SIDE',0.,14)
           CALL NUMBER(-2.05,-4.3,.2,FLOAT(NPB),0.,-1)
           CALL NUMBER(-.25,-4.3,.2,FLOAT(ISB),0.,-1)
           CALL PLOT(0.,0.,-999)
10         CONTINUE
           RETURN
           END
```

APPENDIX 13

SUBROUTINE GPLOT2

```
      SUBROUTINEGPLOT2(NM,NP,X,Y,Z,IA,IB,NPLTS,PCN,IPL,
     2 NWR,NPLTM,NAT,WV,ICN,NCNRS)
      DIMENSIONX(1),Y(1),Z(1),IA(1),IB(1),PCN(3,ICN,IPL)
      DIMENSIONNCNRS(1)
      XNTOT=NWR+NPLTM+NAT
      XNWR=NWR
      XNPLTM=NPLTM
      XNAT=NAT
      DMX=-1.0
      XMN=1.0E10
      YMN=XMN
      ZMN=XMN
      DO110I=1,NP
      IF(X(I).LT.XMN)XMN=X(I)
      IF(Y(I).LT.YMN)YMN=Y(I)
      IF(Z(I).LT.ZMN)ZMN=Z(I)
      DO110J=I,NP
      DIJ=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2+(Z(I)-Z(J))**2)
      IF(DIJ.GT.DMX)DMX=DIJ
110   CONTINUE
      DO160J=1,NPLTS
      IF(NPLTS.EQ.0)GOTO160
      DO390NC=1,NCNRS(J)
      XN=PCN(1,NC,J)
      YN=PCN(2,NC,J)
      ZN=PCN(3,NC,J)
      IF(XN.LT.XMN)XMN=XN
      IF(YN.LT.YMN)YMN=YN
      IF(ZN.LT.ZMN)ZMN=ZN
      DO410I=1,NP
      IF(NP.EQ.0)GOTO410
      DIJ=SQRT((X(I)-XN)**2+(Y(I)-YN)**2+(Z(I)-ZN)**2)
      IF(DIJ.GT.DMX)DMX=DIJ
410   CONTINUE
      DO420K=J,NPLTS
      DO430MC=1,NCNRS(K)
      XM=PCN(1,MC,K)
      YM=PCN(2,MC,K)
      ZM=PCN(3,MC,K)
      DIJ=SQRT((XM-XN)**2+(YM-YN)**2+(ZM-ZN)**2)
```

```
        IF(DIJ.GT.DMX)DMX=DIJ
430     CONTINUE
420     CONTINUE
390     CONTINUE
160     CONTINUE
        F=3.0/DMX
        FW=1.0/(F*WV)
        DO100IV=1,4
        IF(IV.EQ.1)CALLPLOT(0.75,1.5,-3)
        IF(IV.EQ.2)CALLPLOT(6.5,0.0,-3)
        IF(IV.EQ.3)CALLPLOT(-6.5,7.5,-3)
        IF(IV.EQ.4)CALLPLOT(3.5,-3.0,-3)
        IF(IV.EQ.1)CALLSYMBOL(0.0,-0.5,0.2,11HX AXIS VIEW,0.0,11)
        IF(IV.EQ.2)CALLSYMBOL(-3.0,-0.5,0.2,11HY AXIS VIEW,0.0,11)
        IF(IV.EQ.3)CALLSYMBOL(0.0,-3.5,0.2,11HZ AXIS VIEW,0.0,11)
        IF(IV.LE.3)GOTO400
        CALLNUMBER(0.0,1.6,0.2,XNWR,0.0,-1)
        CALLSYMBOL(0.8,1.6,0.2,10HWIRE MODES,0.0,10)
        CALLNUMBER(0.0,1.2,0.2,XNPLTM,0.0,-1)
        CALLSYMBOL(0.8,1.2,0.2,11HPLATE MODES,0.0,11)
        CALLNUMBER(0.0,0.8,0.2,XNAT,0.0,-1)
        CALLSYMBOL(0.8,0.8,0.2,13HATTACH. MODES,0.0,13)
        CALLNUMBER(0.0,0.4,0.2,XNTOT,0.0,-1)
        CALLSYMBOL(0.8,0.4,0.2,11HTOTAL MODES,0.0,11)
        CALLPLOT(0.0,0.1,3)
        CALLPLOT(0.0,-0.1,2)
        CALLPLOT(0.0,0.0,3)
        CALLPLOT(1.0,0.0,2)
        CALLPLOT(1.0,0.1,3)
        CALLPLOT(1.0,-0.1,2)
        CALLSYMBOL(0.05,0.05,0.18,5HSCALE,0.0,5)
        CALLSYMBOL(1.2,0.0,0.2,61,0.0,-1)
        CALLNUMBER(1.6,0.0,0.2,FW,0.0,2)
        CALLSYMBOL(2.6,0.0,0.2,105,0.0,-1)
        GOTO100
400     CONTINUE
        IF(NM.LE.0)GOTO210
        DO200I=1,NM
        N1=IA(I)
        N2=IB(I)
        X1=X(N1)
        Y1=Y(N1)
        Z1=Z(N1)
        X2=X(N2)
        Y2=Y(N2)
        Z2=Z(N2)
        IF(IV.EQ.2)GOTO120
        IF(IV.EQ.3)GOTO130
        XP1=(Y1-YMN)*F
        YP1=(Z1-ZMN)*F
        XP2=(Y2-YMN)*F
        YP2=(Z2-ZMN)*F
        GOTO140
120     CONTINUE
        XP1=-(X1-XMN)*F
        YP1=(Z1-ZMN)*F
        XP2=-(X2-XMN)*F
        YP2=(Z2-ZMN)*F
        GOTO140
130     CONTINUE
        XP1=(Y1-YMN)*F
        YP1=-(X1-XMN)*F
        XP2=(Y2-YMN)*F
        YP2=-(X2-XMN)*F
140     CONTINUE
300     FORMAT(1X,4E15.3)
        CALLSYMBOL(XP1,YP1,0.05,1,0.0,-1)
        CALLPLOT(XP2,YP2,2)
        CALLSYMBOL(XP2,YP2,0.05,1,0.0,-1)
200     CONTINUE
210     CONTINUE
        IF(NPLTS.LE.0)GOTO250
        DO220J=1,NPLTS
        KK=NCNRS(J)+1
        DO230K=1,KK
        L=K
        IF(K.EQ.KK)L=1
        X1=PCN(1,L,J)
        Y1=PCN(2,L,J)
        Z1=PCN(3,L,J)
        IU=2
        IF(K.EQ.1)IU=3
        IF(IV.EQ.2)GOTO270
        IF(IV.EQ.3)GOTO280
        XP=(Y1-YMN)*F
        YP=(Z1-ZMN)*F
        GOTO290
270     CONTINUE
        XP=-(X1-XMN)*F
        YP=(Z1-ZMN)*F
        GOTO290
280     CONTINUE
        XP=(Y1-YMN)*F
        YP=-(X1-XMN)*F
290     CONTINUE
        CALLPLOT(XP,YP,IU)
230     CONTINUE
220     CONTINUE
250     CONTINUE
100     CONTINUE
        CALLPLOT(0.,0.,999)
        RETURN
        END
```

APPENDIX 14

SUBROUTINE ZTOT

```
      SUBROUTINE ZTOT(IA,IB,INM,ISC,I1,I2,I3,JA,JB,MD,NWR,ND,NM,
     2NP,CGD,SGD,D,X,Y,Z,ZLD,NPL,NAT,ZS,IRDZM,ZLDA,PA,PB,
     3NSA,NPLA,PCN,IPL,IPLM,BDSK,ZT,ZTF,NM12N,NM23N,ICN,
     4NDNPLT,NOVT,INT,INTP,INTD,CMM,ERVSR,RMIN,DR,IAT,IPN,
     5IQUAD,NCNRS,IFIL,IREC,ICC)
      DIMENSIONIA(1),IB(1),ISC(1),I1(1),I2(1),I3(1),JA(1),JB(1),
     2MD(INM,4),ND(1),D(1),X(1),Y(1),Z(1),IPN(1),DR1(6)
      DIMENSIONPA(IPLM,4,3),PB(IPLM,4,3),NSA(1),NPLA(1),PCN(3,ICN,IPL)
      DIMENSION BDSK(1),NM12N(1),NM23N(1),NDNPLT(1),RMIN(1),DR(1)
      DIMENSION PDIST(6),IQUAD(1),DNOM1(1),NCNRS(1),IREC(1)
      COMPLEXZLDA(1),SGD(1),CGD(1),ZLD(1),ZTF(ICC,ICC),ZT(1)
      COMPLEXZS,XJ,ETA,GAM,ZMN,EP3,ERVSR(IAT,400),EX,EY,EZ,DZMN
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
      AB(I,A,B)=(I-1)*B+(2-I)*A
      IJ(I,J,NT)=(J-1)*NT-(J*J-J)/2+I
      IWG=0
      IWZ=0
      ICAL=1
      IF(IWG+IWZ.GT.0)WRITE(6,365)IRDZM
  365 FORMAT(//3X,'Z MATRIX READ OPTION: IRDZM = ',I3/)
      IF(IWG.EQ.1)WRITE(6,360)NWR,NPL,NAT
      E0=8.85E-12
      EP3=CMPLX(E0,0.0)
      NTOT=NWR+NDNPLT(NPL)+NOVT+NAT
      WRITE(6,*)NTOT,NWR,NDNPLT(NPL),NOVT,NAT
      NDT=18
      NDE=18
      IFGD=0
C
C     EVALUATE THE IMPEDANCE ARRAY ZT(MN) (IF IFIL = 0) OR
C     IMPEDANCE MATRIX ZTF(M,N) (IF IFIL = 1)
C
C (COUNTING ACROSS)
C REGION TEST EXP
C  1       W    W
C  2       W    P
C  3       W    A
C  4       P    W
C  5       P    P
C  6       P    A
```

```
C  7      A    W
C  8      A    P
C  9      A    A
C
C     CALCULATE REGION 1 (WIRE/WIRE).
C
        IF(NWR.EQ.0)GOTO100
        IF(IRDZM.EQ.1)GOTO100
        FHZ=3.E8/WV
        AEFAC=1.0
        AEQ=AEFAC*A
        CALLSGANT(IA,IB,INM,INT,ISC,I1,I2,I3,JA,JB,MD,NWR,ND,NM,NP,AEQ,
     2  A,ZT,CGD,CMM,D,ETA,EP3,ETA,FHZ,GAM,SGD,X,Y,Z,ZLD,ZS)
        DO110J=1,NWR
        JJ=NWR-J+1
        DO110I=1,J
        II=NWR-I+1
C**************************
C Regular surface patch test modes are used (IFIL = 0). Then
C the impedances calculated by SGANT are stored in the one
C dimensional array ZT.
C**************************
        IF(IFIL.EQ.0)THEN
        IJN=IJ(II,JJ,NTOT)
        K=IJ(II,JJ,NWR)
        ZT(IJN)=ZT(K)
        END IF
        IF(IFIL.EQ.1)THEN
C**************************
C Filamentary surface patch test modes are used (IFIL = 1). Then
C the impedances calculated by SGANT are stored in the two
C dimensional array ZTF.
C**************************
        K=IJ(II,JJ,NWR)
        ZTF(JJ,II)=ZT(K)
        ZTF(II,JJ)=ZTF(JJ,II)
        END IF
        IF(IWZ.EQ.1)WRITE(6,390)I,J,IJN,ZT(IJN)
390     FORMAT(2X,2I4,I6,2E15.5)
110     CONTINUE
100     CONTINUE
        IF(NPL+NAT.EQ.0)RETURN
C
C     CALCULATE REGIONS 2 THROUGH 9.
C
      DO130N=1,NTOT
        IF(IFIL.EQ.0)NN=N
        IF(IFIL.EQ.1)NN=1
C**************************
C If regular surface patch test modes are used (IFIL = 0) the
C resulting impedance matrix is symmetric; i.e. only the lower
C triangular part of it is calculated and the entries are stored
C in array ZT. If filamentary surface patch modes are used (IFIL = 1)
C the whole impedance matrix is calculated and stored in array
C ZTF(M,N).
C***************************
      DO140M=NN,NTOT
      INDI=0
      IF(IFIL.EQ.0)MN=IJ(M,N,NTOT)
      IF(M.LE.NWR .AND. N.LE.NWR)GOTO 140
      IF(M.LE.NWR) GOTO 115
      IF(N.LE.NWR) GOTO 115
      IF(N.GT.NWR+NDNPLT(NPL))GO TO 115
      IF(M.GT.NWR+NDNPLT(NPL))GO TO 115
C**************************
C Determine what, if any, plate we are on.
C**************************
      DO 20 I=1,NPL
      IF (M-NWR .GT. NDNPLT(I)) GOTO 20
      GOTO 25
20    CONTINUE
      GOTO 115
25    DO 30 J=1,NPL
      IF (N-NWR .GT. NDNPLT(J)) GOTO 30
      GOTO 35
30    CONTINUE
      GOTO 115
35    IF (I .NE. J) GOTO 115
C**************************
C We are on plate I=J
C**************************
      IF(IREC(I).EQ.0)GO TO 115
C*************************
C Calculation of the mutual impedance between two
C modes on the same rectangular plate using the
C TOEPLITZ properties.
C*************************
      NM12=NM12N(I)
      NM23=NM23N(I)
      NDNI=0
      IF (I .GT. 1) NDNI=NDNPLT(I-1)
      IF(IPN(I).EQ.2)NDNI=NDNI-(NM12-1)*NM23
      K=M-NWR-NDNI
      L=N-NWR-NDNI
      IF(K.LT.L)GO TO 140
      IF (L .EQ. 1 .OR. L .EQ. NM23*(NM12-1)+1)THEN
      INDI=1
      GO TO 115
      END IF
      CALL TOPO(NM12,NM23,K,L,MT,NT,SGN)
      K=MT+NWR+NDNI
      L=NT+NWR+NDNI
C**********************
C Regular surface patch test modes are used (IFIL = 0).
C**********************
      IF(IFIL.EQ.0)THEN
      KL=IJ(K,L,NTOT)
      ZT(MN)=ZT(KL)*SGN
      END IF
C***********************
C Filamentary surface patch test modes are used (IFIL = 1).
C***********************
      IF(IFIL.EQ.1)THEN
      ZTF(M,N)=ZTF(K,L)*SGN
      ZTF(N,M)=ZTF(M,N)
      END IF
      GOTO 140
115   DO150I=1,2
      DO160J=1,2
C**********+*************
C The polarity indicators IM12 and IN12 are defined
C below.
C************************
      IM12=(-1)**I
      IN12=(-1)**J
C
C     DETERMINE TEST MODE TYPE.
C
      IF(M.GT.NWR+NDNPLT(NPL)+NOVT) GOTO 190
      IF(M.GT.NWR) GOTO 180
C
C     TEST MODE IS A WIRE
C
C***************************
C The geometry of the wire test mode is
C defined below.
C***************************
      K=M
      I1K=I2(K)
      I2K=I1(K)
      IF(I.EQ.2)I2K=I3(K)
      KWS=JA(K)
      IF(I.EQ.2)KWS=JB(K)
      XM1=X(I1K)
      YM1=Y(I1K)
      ZM1=Z(I1K)
      XM2=X(I2K)
      YM2=Y(I2K)
      ZM2=Z(I2K)
      IOP=3
      IIOP=0
      IF(IWG.EQ.1)WRITE(6,360)M,N,K,IOP,IM12,XM1,YM1,ZM1,XM2,YM2,ZM2
360   FORMAT(2X,5I4,12F8.3)
      GOTO280
180   CONTINUE
C
C     TEST MODE IS A PLATE.
C
      K=M-NWR
C********************************
C The geometry of the test plate mode is defined
C below.
C********************************
      IF(IFIL.EQ.0)IOP=1
      IF(IFIL.EQ.1)IIOP=3
      IACM=IQUAD(K)
      XM1=AB(I,PA(K,1,1),PB(K,1,1))
      YM1=AB(I,PA(K,1,2),PB(K,1,2))
      ZM1=AB(I,PA(K,1,3),PB(K,1,3))
      XM2=AB(I,PA(K,2,1),PB(K,2,1))
      YM2=AB(I,PA(K,2,2),PB(K,2,2))
      ZM2=AB(I,PA(K,2,3),PB(K,2,3))
      XM3=AB(I,PA(K,3,1),PB(K,3,1))
      YM3=AB(I,PA(K,3,2),PB(K,3,2))
      ZM3=AB(I,PA(K,3,3),PB(K,3,3))
      XM4=AB(I,PA(K,4,1),PB(K,4,1))
      YM4=AB(I,PA(K,4,2),PB(K,4,2))
      ZM4=AB(I,PA(K,4,3),PB(K,4,3))
C*************************************
C If IFIL = 1 the endpoints of the filamentary
C test surface patch monopole are defined
C below.
C************************************
        IF(IFIL.EQ.1)THEN
        XM1=(XM1+XM4)/2.0
        YM1=(YM1+YM4)/2.0
        ZM1=(ZM1+ZM4)/2.0
        XM2=(XM2+XM3)/2.0
        YM2=(YM2+YM3)/2.0
        ZM2=(ZM2+ZM3)/2.0
        GO TO 280
        END IF
        IF(IWG.EQ.1)WRITE(6,360)M,N,K,IOP,IM12,XM1,YM1,ZM1,XM2,YM2,ZM2,
     &  XM3,YM3,ZM3,XM4,YM4,ZM4
        GOTO280
190     CONTINUE
C
C     TEST MODE IS AN ATTACHMENT MODE.
C
      K=M-NWR-NDNPLT(NPL)-NOVT
      IM12=1
      IF(I.EQ.2)GOTO200
C*******************************
C The geometry of the disk monopole of the attachment
C test mode is defined below.
C*******************************
      IOP=2
      IIOP=0
      NAS=NSA(K)
      IF(NAS.GT.NM)GOTO210
      NAP=IA(NAS)
      GOTO220
  210 CONTINUE
      NAS=NAS-NM
      NAP=IB(NAS)
  220 CONTINUE
      BM=BDSK(K)
      XM1=X(NAP)
      YM1=Y(NAP)
      ZM1=Z(NAP)
      NPLK=NPLA(K)
      XM2=PCN(1,1,NPLK)
      YM2=PCN(2,1,NPLK)
      ZM2=PCN(3,1,NPLK)
      XM3=PCN(1,2,NPLK)
      YM3=PCN(2,2,NPLK)
      ZM3=PCN(3,2,NPLK)
      IF(IWG.EQ.1)WRITE(6,360)M,N,K,IOP,IM12,XM1,YM1,ZM1,XM2,YM2,ZM2,
     2XM3,YM3,ZM3
      GOTO280
  200 CONTINUE
C*******************************
```

```
C The geometry of the wire monopole of the
C attachment test mode is defined below.
C*****************************
      IOP=3
      IIOP=0
      NAS=NSA(K)
      IF(NAS.GT.NM)GOTO230
      NPP=IB(NAS)
      GOTO240
  230 CONTINUE
      NAS=NAS-NM
      NPP=IA(NAS)
  240 CONTINUE
      XM2=X(NPP)
      YM2=Y(NPP)
      ZM2=Z(NPP)
      IF(IWG.EQ.1)WRITE(6,360)M,N,K,IOP,IM12,XM1,YM1,ZM1,XM2,YM2,ZM2
      GOTO 280
  280 CONTINUE
C
C     DETERMINE EXPANSION MODE TYPE.
C
      IF(N.GT.NWR+NDNPLT(NPL)+NOVT) GOTO 270
      IF(N.GT.NWR) GOTO 260
C
C     EXPANSION MODE IS A WIRE.
C
      L=N
      JOP=3
      IF(IRDZM.NE.1)GOTO262
      IF(IOP.GE.2.AND.JOP.GE.2)GOTO140
262   IF(IRDZM.NE.2)GOTO263
      IF(IOP.EQ.1.AND.JOP.EQ.1)GOTO140
263   CONTINUE
C****************************
C The geometry of the expansion wire monopole
C is defined below.
C****************************
      IF(I+J.EQ.2)THEN
      IF(IFIL.EQ.0)ZT(MN)=(0.0,0.0)
      IF(IFIL.EQ.1)ZTF(M,N)=(0.0,0.0)
      END IF
      I1L=I2(L)
      I2L=I1(L)
      IF(J.EQ.2)I2L=I3(L)
      XN1=X(I1L)
      YN1=Y(I1L)
      ZN1=Z(I1L)
      XN2=X(I2L)
      YN2=Y(I2L)
      ZN2=Z(I2L)
      IF(IWG.EQ.1)WRITE(6,360)M,N,L,JOP,IN12,XN1,YN1,ZN1,XN2,YN2,ZN2
261   IF (ICAL.NE.1) GO TO 6001
      IF(IIOP.EQ.3)IOP=3
C*****************************
C Test monopole is a wire.
C*****************************
      IF(IOP.EQ.3) CALL ZWTWE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,
     &XN1,YN1,ZN1,XN2,YN2,ZN2,IN12,ZMN,1)
C*******************************
C Test monopole is a disk.
C*******************************
      IF(IOP.EQ.2)CALLDSKTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,
     &3,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,0,
     &INTD,BM,BN,ZMN)
      IF(IOP.NE.1)GO TO 6001
C*******************************
C Test monopole is a surface patch.
C*******************************
        XMDN=(XN1+XN2)/2.
        YMDN=(YN1+YN2)/2.
        ZMDN=(ZN1+ZN2)/2.
        XMDM=(XM1+XM2+XM3+XM4)/4.
        YMDM=(YM1+YM2+YM3+YM4)/4.
        ZMDM=(ZM1+ZM2+ZM3+ZM4)/4.
        DISMN=SQRT((XMDM-XMDN)**2+(YMDM-YMDN)**2+(ZMDM-ZMDN)**2)
        IF(DISMN.LE.0.25*WV)NPT=8
        IF(DISMN.GT.0.25*WV.AND.DISMN.LE.0.35*WV)NPT=4
        IF(DISMN.GT.0.35*WV)NPT=2
        IF(IACM.NE.-3)CALL PLTST2(XM4,YM4,ZM4,XM1,YM1,ZM1,XM2,
     1  YM2,ZM2,XM3,YM3,ZM3,IM12,JOP,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,
     2  XN4,YN4,ZN4,IN12,NPT,0,BN,IACM,IACN,ZMN)
        IF(IACM.EQ.-3)CALL PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,
     &  YM3,ZM3,IM12,3,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,NPT,
     &  0,BN,ZMN)
6001    IF(IFIL.EQ.1)ZTF(M,N)=ZTF(M,N)+ZMN
        IF(IFIL.EQ.0)ZT(MN)=ZT(MN)+ZMN
        GOTO160

  260 CONTINUE
C
C     EXPANSION MODE IS A PLATE.
C
      JOP=1
      IF(IRDZM.NE.1)GO TO 264
      IF(IOP.GE.2.AND.JOP.GE.2)GO TO 140
264   IF(IRDZM.NE.2)GO TO 265
      IF(IIOP.EQ.3)GO TO 140
      IF(IOP.EQ.1.AND.JOP.EQ.1)GO TO 140
265   CONTINUE
      L=N-NWR
C*******************************
C The geometry of the expansion surface patch
C monopole is defined below.
C*******************************
      IACN=IQUAD(L)
      XN1=AB(J,PA(L,1,1),PB(L,1,1))
      YN1=AB(J,PA(L,1,2),PB(L,1,2))
      ZN1=AB(J,PA(L,1,3),PB(L,1,3))
      XN2=AB(J,PA(L,2,1),PB(L,2,1))
      YN2=AB(J,PA(L,2,2),PB(L,2,2))
      ZN2=AB(J,PA(L,2,3),PB(L,2,3))
      XN3=AB(J,PA(L,3,1),PB(L,3,1))
      YN3=AB(J,PA(L,3,2),PB(L,3,2))
      ZN3=AB(J,PA(L,3,3),PB(L,3,3))
      XN4=AB(J,PA(L,4,1),PB(L,4,1))
      YN4=AB(J,PA(L,4,2),PB(L,4,2))
      ZN4=AB(J,PA(L,4,3),PB(L,4,3))
      IF(IIOP.EQ.3)IOP=3
      IF(I+J.EQ.2)THEN
      IF(IFIL.EQ.0)ZT(MN)=(0.0,0.0)
      IF(IFIL.EQ.1)ZTF(M,N)=(0.0,0.0)
      END IF
      IF(IWG.EQ.1)WRITE(6,360)M,N,L,JOP,IN12,XN1,YN1,ZN1,XN2,YN2,ZN2,
     1XN3,YN3,ZN3
      IF (ICAL .NE. 1) GOTO 6002
      XMDN=(XN1+XN2+XN3+XN4)/4.0
      YMDN=(YN1+YN2+YN3+YN4)/4.0
      ZMDN=(ZN1+ZN2+ZN3+ZN4)/4.0
      IF(IOP.NE.1)GO TO 431
C************************
C Test monopole is a surface patch.
C************************
        XMDM=(XM1+XM2+XM3+XM4)/4.0
        YMDM=(YM1+YM2+YM3+YM4)/4.0
        ZMDM=(ZM1+ZM2+ZM3+ZM4)/4.0
        DISMN=SQRT((XMDM-XMDN)**2+(YMDM-YMDN)**2+(ZMDM-ZMDN)**2)
        IF(DISMN.LE.0.25*WV)NPT=8
        IF(DISMN.GT.0.25*WV.AND.DISMN.LE.0.35*WV)NPT=4
        IF(DISMN.GT.0.35*WV)NPT=2
        IF(IACN.NE.-3.AND.DISMN.GE.0.6*WV)NPT=1
        IF(IACM.EQ.-3.AND.IACN.EQ.-3)GO TO 492
        GO TO 479
  492   CONTINUE
        CALL PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,1,
     &  XN1,YN1,ZN1,XN2.YN2,ZN2.XN3,YN3,ZN3,IN12,NPT,NPT,BN,
     &  ZMN)
        GO TO 6002
479     CONTINUE
        IF(IOP.EQ.1)CALL PLTST2(XM4,YM4,ZM4,
     &  XM1,YM1,ZM1,XM2,YM2,ZM2,
     &  XM3,YM3,ZM3,IM12,JOP,XN4,YN4,ZN4,XN1,YN1,ZN1,XN2,YN2,
     &  ZN2,XN3,YN3,ZN3,IN12,NPT,NPT,BN,IACM,IACN,ZMN)
        GO TO 6002
431     CONTINUE
        IF(IOP.NE.2)GO TO 458
C**************************
C Test monopole is a disk
C**************************
      IF(L.GT.NDNPLT(NPL))GO TO 5100
C**************************
C CHECK FOR PARALLEL PLATE-DISK
C**************************
      PNX=(YN2-YN1)*(ZN3-ZN2)-(YN3-YN2)*(ZN2-ZN1)
      PNY=(XN3-XN2)*(ZN2-ZN1)-(XN2-XN1)*(ZN3-ZN2)
      PNZ=(XN2-XN1)*(YN3-YN2)-(XN3-XN2)*(YN2-YN1)
      DN12=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
      DN23=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
      DNX=(YM2-YM1)*(ZM3-ZM2)-(YM3-YM2)*(ZM2-ZM1)
      DNY=(XM3-XM2)*(ZM2-ZM1)-(XM2-XM1)*(ZM3-ZM2)
      DNZ=(XM2-XM1)*(YM3-YM2)-(XM3-XM2)*(YM2-YM1)
      DM12=SQRT((XM2-XM1)**2+(YM2-YM1)**2+(ZM2-ZM1)**2)
      DM23=SQRT((XM3-XM2)**2+(YM3-YM2)**2+(ZM3-ZM2)**2)
      COSTH=(PNX*DNX+PNY*DNY+PNZ*DNZ)/SQRT((PNX**2+
     &PNY**2+PNZ**2)*(DNX**2+DNY**2+DNZ**2))
      IF(ABS(COSTH).LT. .997) GOTO 5100
C********************************
C PLATE AND DISK ARE PARALLEL
C CHECK FOR FIRST COLUMN OF PLATE
C********************************
      IF(J.EQ.2) GOTO 5005
      KPL=1
      IF(L.EQ.1) GOTO 5002
      DO 5001 II=2,NPL
      KPL=II
      IF(L.EQ.NDNPLT(II-1)+1) GOTO 5002
5001  CONTINUE
C**************************
C NOT ON A FIRST PLATE COLUMN
C**************************
5005  RMINK=RMIN(K)
      DRK=DR(K)
      DIST=PDIST(K)
      IF(IACN.EQ.-3)CALL PDPZ(XM1,YM1,ZM1,K,XN1,YN1,ZN1,XN2,YN2,ZN2,
     &XN3,YN3,ZN3,IN12,INTP,ERVSR,IAT,RMINK,DRK,ZMN,DIST)
      IF(IACN.NE.-3)CALL PDPZ1(XM1,YM1,ZM1,K,XN4,YN4,ZN4,XN1,YN1,ZN1,
     &XN2,YN2,ZN2,XN3,YN3,ZN3,IACN,IN12,INTP,ERVSR,IAT,RMINK,DRK,ZMN,
     &DIST)
      GOTO 6002
C*************************
C ON A FIRST PLATE COLUMN
C FIND RMAX,RMIN
C**************************
5002  IF(IREC(KPL).EQ.0)GO TO 5003
      PX0=(PCN(1,1,KPL)+PCN(1,3,KPL))/2.
      PY0=(PCN(2,1,KPL)+PCN(2,3,KPL))/2.
      PZ0=(PCN(3,1,KPL)+PCN(3,3,KPL))/2.
      DIAG=.5*SQRT((PCN(1,1,KPL)-PCN(1,3,KPL))**2+
     &(PCN(2,1,KPL)-PCN(2,3,KPL))**2+(PCN(3,1,KPL)-PCN(3,3,KPL))**2)
      R=SQRT((PX0-XM1)**2+(PY0-YM1)**2+(PZ0-ZM1)**2)
      RMAX=BDSK(K)+R+DIAG
      RMIN(K)=R-BDSK(K)-DIAG
      IF(RMIN(K).LT.0.) RMIN(K)=0.
      GO TO 5006
5003  RMIN(K)=0.0
      DIAG=0.0
      PX0=0.0
      PY0=0.0
      PZ0=0.0
      DO 5004 IKC=1,NCNRS(KPL)
      PX0=PX0+PCN(1,IKC,KPL)
      PY0=PY0+PCN(2,IKC,KPL)
      PZ0=PZ0+PCN(3,IKC,KPL)
      DCC=SQRT((XM1-PCN(1,IKC,KPL))**2+(YM1-PCN(2,IKC,KPL))**2
     1+(ZM1-PCN(3,IKC,KPL))**2)
      IF(DCC.GT.DIAG)DIAG=DCC
5004  CONTINUE
      PX0=PX0/NCNRS(KPL)
      PY0=PY0/NCNRS(KPL)
      PZ0=PZ0/NCNRS(KPL)
      R=SQRT((PX0-XM1)**2+(PY0-YM1)**2+(PZ0-ZM1)**2)
      RMAX=R+DIAG+BDSK(K)
```

```
5006   NINA=2*(60.*(RMAX-RMIN(K))/WV)
       NINA1=NINA+1
       IF(NINA1 .GT. 400) WRITE(6,5111) M,N,NINA1
5111   FORMAT('ERROR-NINA GT 400',3I5)
       DR(K)=(RMAX-RMIN(K))/NINA
C*******************************
C Compute the distance between the plane of
C the disk monopole and the plane of the surface
C patch.
C*******************************
       PND=-PNX*XN1-PNY*YN1-PNZ*ZN1
       DIST=(PNX*XM1+PNY*YM1+PNZ*ZM1+PND)/SQRT(PNX**2+PNY**2+PNZ**2)
       DIST=ABS(DIST)
       IF(DIST.LT.Q) DIST=Q
       PDIST(K)=DIST
C*****************************
C FILL ARRAY ZVSR
C*****************************
       DO 5010 JJ=1,NINA1
       R=RMIN(K)+(JJ-1)*DR(K)
       CALL ERDSK(A,BDSK(K),R,DIST,ETA,WV,100,EX)
5010   ERVSR(K,JJ)=EX
       GOTO 5005
C******************************
C Disk monopole is not parallel to the surface
C patch.
C******************************
5100   DISMN=SQRT((XM1-XMDN)**2+(YM1-YMDN)**2+(ZM1-ZMDN)**2)
       IF(DISMN.LE.0.35*WV)THEN
       INTP=4
       NDT=8
       END IF
       IF(DISMN.GT.0.35*WV.AND.DISMN.LE.0.6*WV)THEN
       INTP=4
       NDT=4
       END IF
       IF(DISMN.GT.0.6*WV)THEN
       INTP=2
       NDT=4
       END IF
       IF(IACN.NE.-3)CALL DSKTS2(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,
      1JOP,XN4,YN4,ZN4,XN1.YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,
      2NDT,BM,BN,ZMN)
       IF(IACN.EQ.-3)CALL DSKTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,
      &1,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,NDT,
      &BM,BN,ZMN)
       GO TO 6002
458    IF(IOP.NE.3)GO TO 453
C*******************************
C Expansion monopole is a wire.
C*******************************
         KINT=0
         DM=SQRT((XM2-XM1)**2+(YM2-YM1)**2+(ZM2-ZM1)**2)
         XMDM=(XM1+XM2)/2.
         YMDM=(YM1+YM2)/2.
         ZMDM=(ZM1+ZM2)/2.
         DISMN=SQRT((XMDM-XMDN)**2+(YMDM-YMDN)**2+(ZMDM-ZMDN)**2)
         IF(DISMN.LE.0.25*WV)NPT=8
```

```
        IF(DISMN.GT.0.25*WV.AND.DISMN.LE.0.35*WV)NPT=4
        IF(DISMN.GT.0.35*WV)NPT=2
        IF(IACN.NE.-3.AND.DISMN.GE.0.6*WV)NPT=1
 453    CONTINUE
        IF(IOP.EQ.3.AND.IACN.NE.-3)CALL ZWTPE2(XM1,YM1,ZM1,XM2,YM2,ZM2,DM,IOP,
     2  IM12,XN4,YN4,ZN4,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,
     3  NPT,IACN,ZMN,KINT)
        IF(IOP.EQ.3.AND.IACN.EQ.-3)CALL ZWTPE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,
     1  XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,NPT,ZMN)
        IF(IFGD.EQ.1.AND.IOP.EQ.3)CALL PLOT(.0,.0,999)
        IF(IFGD.GT.0.AND.IOP.EQ.3)WRITE(6,*)NPE,ZMN
C*****************************
C Entry MN of array ZT(MN) (if IFIL = 0) or entry
C (M,N) of array ZTF(M,N) (if IFIl = 1) is defined
C below.
C******************************
6002    IF(IFIL.EQ.1)ZTF(M,N)=ZTF(M,N)+ZMN
        IF(IFIL.EQ.1.AND.INDI.EQ.1)ZTF(N,M)=ZTF(M,N)
        IF(IFIL.EQ.0)ZT(MN)=ZT(MN)+ZMN
        IF(IWZ.EQ.1)WRITE(6,*)M,N,ZT(MN),ZMN
        IF(IWZ.EQ.1)WRITE(6,410)M,N,I,J,1,NPT,NPE,CPU
410     FORMAT(1X,' M = ',I3,2X,' N = ',I3,2X,' I = ',I2,2X,
     1  ' J = ',I2,2X,I1,' NPT = ',I2,' NPE = ',I2,2X,
     2  ' CPU = ',F12.6)
        GO TO 160
270     CONTINUE
C***************************
C     EXPANSION MODE IS AN ATTACHMENT MODE.
C***************************
        JOP=2
        IF(IRDZM.NE.1)GOTO266
        IF(IOP.GE.2.AND.JOP.GE.2)GOTO140
  266   IF(IRDZM.NE.2)GOTO267
        IF(IOP.EQ.1.AND.JOP.EQ.1)GOTO140
  267   CONTINUE
        IF(I+J.EQ.2)THEN
        IF(IFIL.EQ.0)ZT(MN)=(0.0,0.0)
        IF(IFIL.EQ.1)ZTF(M,N)=(0.0,0.0)
        END IF
        L=N-NWR-NDNPLT(NPL)-NOVT
        IN12=1
        IF(J.EQ.2)GOTO290
C***************************
C The geometry of the disk monopole of the expansion
C attachment mode is defined below.
C***************************
      NAS=NSA(L)
      IF(NAS.GT.NM)GOTO300
      NAP=IA(NAS)
      NAP2=IB(NAS)
      GOTO310
  300 CONTINUE
      NAS=NAS-NM
      NAP=IB(NAS)
      NAP2=IA(NAS)
  310 CONTINUE
      BN=BDSK(L)
      XN1=X(NAP)
      YN1=Y(NAP)
      ZN1=Z(NAP)
      XNT=X(NAP2)
      YNT=Y(NAP2)
      ZNT=Z(NAP2)
      NPLL=NPLA(L)
      XN2=PCN(1,1,NPLL)
      YN2=PCN(2,1,NPLL)
      ZN2=PCN(3,1,NPLL)
      XN3=PCN(1,2,NPLL)
      YN3=PCN(2,2,NPLL)
      ZN3=PCN(3,2,NPLL)
      IF(IWG.EQ.1)WRITE(6,360)M,N,L,JOP,IN12,XN1,YN1,ZN1,XN2,YN2,ZN2,
     2XN3,YN3,ZN3
      GOTO 335
  290 CONTINUE
C************************
C The wire monopole of the attachment mode is
C defined below.
C************************
      NAS=NSA(L)
      IF(NAS.GT.NM)GOTO320
      NPP=IB(NAS)
      GOTO330
  320 CONTINUE
      NAS=NAS-NM
      NPP=IA(NAS)
  330 CONTINUE
      XN2=X(NPP)
      YN2=Y(NPP)
      ZN2=Z(NPP)
      JOP=3
      IF(IOP.NE.3.OR.JOP.NE.3)GOTO341
      IF(NAS.NE.KWS)GOTO341
      IF(I1(K).EQ.NAP)GOTO342
      DZMN=(GAM*D(NAS)*CGD(NAS)-SGD(NAS))*ZS/(4.*PI*GAM*A*SGD(NAS)**2)
      GOTO343
342   DZMN=(SGD(NAS)*CGD(NAS)-GAM*D(NAS))*ZS/(4.*PI*GAM*A*SGD(NAS)**2)
343   ZMN=ZMN+DZMN*IM12*IN12
341   CONTINUE
      IF(IWG.EQ.1)WRITE(6,360)M,N,L,JOP,IN12,XN1,YN1,ZN1,XN2,YN2,ZN2
340   IF (ICAL .NE. 1) GOTO 336
C***************************
C Test monopole is a wire.
C***************************
      IF(IOP.EQ.3)CALLZWTWE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,
     2XN1,YN1,ZN1,XN2,YN2,ZN2,IN12,ZMN,1)
C**************************
C Test monopole is a disk.
C**************************
      IF(IOP.EQ.2)CALL DSKTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,
     &3,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,0,INTD,
     &BM,BN,ZMN)
C****************************
C Test monopole is a surface patch.
C****************************
      IF(IOP.EQ.1.AND.IACM.EQ.-3)CALL PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,
     &ZM3,IM12,3,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,
     &0,BN,ZMN)
      IF(IOP.EQ.1.AND.IACM.NE.-3)CALL PLTST2(XM4,YM4,ZM4,XM1,YM1,ZM1,XM2,YM2,
     &ZM2,XM3,YM3,ZM3,IM12,JOP,XN4,YN4,ZN4,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,
     &ZN3,IN12,NPT,NDE,BN,IACM,IACN,ZMN)
      GOTO 336
335   IF(IOP .NE. 2) GOTO 531
      IF(K.NE.L)GOTO531
C*****************************
C COMPUTE ATTACHMENT TO ATTACHMENT (SELF) IMPEDANCE
C*****************************
        DWIRE=D(NAS)
        DXW=(XNT-XN1)/DWIRE
        DYW=(YNT-YN1)/DWIRE
        DZW=(ZNT-ZN1)/DWIRE
        DPL=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
        DXP=(XN2-XN1)/DPL
        DYP=(YN2-YN1)/DPL
        DZP=(ZN2-ZN1)/DPL
        DPL=SQRT((XN3-XN1)**2+(YN3-YN1)**2+(ZN3-ZN1)**2)
        DXQ=(XN3-XN1)/DPL
        DYQ=(YN3-YN1)/DPL
        DZQ=(ZN3-ZN1)/DPL
        DXN=DYP*DZQ-DYQ*DZP
        DYN=DXQ*DZP-DXP*DZQ
        DZN=DXP*DYQ-DYP*DXQ
        COSA=DXW*DXN+DYW*DYN+DZW*DZN
        COSA=ABS(0.999*COSA)
        PSI=ACOS(COSA)
        PSI=PSI*180.0/PI
        CALL ZATAT2(BM,DWIRE,ZMN,40,ZS,PSI)
        IF(IFIL.EQ.0)ZT(MN)=ZMN+ZLDA(K)
        IF(IFIL.EQ.1)ZTF(M,N)=ZMN+ZLDA(K)
        IF(IWZ.EQ.1) WRITE(6,370)M,N,ZMN
        GOTO 1073
C******************************
C EXPANSION MONOPOLE IS A DISK.
C******************************
531     IF (ICAL .NE. 1) GOTO 336
C******************************
C Test monopole is a wire.
C******************************
        IF(IOP.EQ.3)CALLZWTDE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,
     &  XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,INTD,BN,ZMN)
C******************************
C Test monopole is a disk.
C******************************
        IF(IOP.EQ.2)CALLDSKTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,
     &  2,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTD,INTD,
     &  BM,BN,ZMN)
C******************************
C Test monopole is a surface patch.
C******************************
       IF(IOP.EQ.1.AND.IACM.EQ.3)CALL PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,
     & ZM3,IM12,3,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,
     & 0,BN,ZMN)
       IF(IOP.EQ.1.AND.IACM.NE.-3)CALL PLTST2(XM4,YM4,ZM4,XM1,YM1,ZM1,XM2,YM2,
     & ZM2,XM3,YM3,ZM3,IM12,JOP,XN4,YN4,ZN4,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,
     & ZN3,IN12,NPT,0,BN,IACM,IACN,ZMN)
```

```
  336   CONTINUE
        IF(IFIL.EQ.0)ZT(MN)=ZT(MN)+ZMN
        IF(IFIL.EQ.1)ZTF(M,N)=ZTF(M,N)+ZMN
        IF(IWZ.EQ.1)WRITE(6,370)M,N,ZT(MN),ZMN
  370   FORMAT(6X,2I4,4E12.3)
        GOTO 160
  160   CONTINUE
  150   CONTINUE
        IF(IWG.EQ.1)WRITE(6,380)
  380   FORMAT(/)
 1073   CONTINUE
 140    CONTINUE
 130    CONTINUE
        CALL GETCP(IIII)
        CPU1=(IIII-III)/100.
 9973   CONTINUE
        RETURN
        END
```

APPENDIX 15

SUBROUTINE TOPO

```
      SUBROUTINE TOPO(NM12,NM23,K,L,MT,NT,SGN)
C CHECK TOEPLITZ PROPERTIES
      SGN=1.
      IF (K .GT. NM23*(NM12-1)) GOTO 20
C X POLARIZATIONS
      NRM=(K-1)/(NM12-1)+1
      NCM=K-(NRM-1)*(NM12-1)
      NRN=(L-1)/(NM12-1)+1
      NCN=L-(NRN-1)*(NM12-1)
      NU=NRM-NRN
      NO=IABS(NCM-NCN)
      MT=NU*(NM12-1)+NO+1
      NT=1
      RETURN
20    IF (L .LE. NM23*(NM12-1)) GOTO 30
C Y POLARIZATIONS
      MM=K-NM23*(NM12-1)
      NN=L-NM23*(NM12-1)
      NRMM=(MM-1)/(NM23-1)+1
      NCMM=MM-(NRMM-1)*(NM23-1)
      NRNN=(NN-1)/(NM23-1)+1
      NCNN=NN-(NRNN-1)*(NM23-1)
      NU=NRMM-NRNN
      NO=IABS(NCMM-NCNN)
      MT=NU*(NM23-1)+NO+1+NM23*(NM12-1)
      NT=1+NM23*(NM12-1)
      RETURN
30    CONTINUE
C XY POLARIZATIONS
      MM=K-NM23*(NM12-1)
      NSOM=2*((MM-1)/(NM23-1))+1
      NSUM=2*(MM-((NSOM-1)/2)*(NM23-1))
      NSUN=2*((L-1)/(NM12-1))+1
      NSON=2*(L-((NSUN-1)/2)*(NM12-1))
      NU=NSUN-NSUM
      NO=NSON-NSOM
      IF (NU .GT. 0) SGN=-SGN
      IF (NU .GT. 0) NU=-NU
      IF (NO .GT. 1) SGN=-SGN
      IF (NO .GT. 1) NO=-NO
      MT=((1-NO)/2)*(NM23-1)+(1-NU)/2+NM23*(NM12-1)
      NT=1
      RETURN
      END
```

APPENDIX 16

SUBROUTINE PLTTST

```
      SUBROUTINE PLTTST(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,
     &JOP,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,NINT,
     &BN,ZMN)
      COMPLEX Z,ZMN.GAM,ETA
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
      IF(JOP.NE.1)GO TO 100
      XN3=XN2
      YN3=YN2
      ZN3=ZN2
  100 CONTINUE
      D=SQRT((XM1+XM3-XN1-XN3)**2+(YM1+YM3-YN1-YN3)**2+
     &(ZM1+ZM3-ZN1-ZN3)**2)/2.
      NPT=INTP
      IF(D.GT. .25*WV)NPT=2*(INTP/6)
      IF(NPT.LT.2)NPT=2
      ZMN=(0.,0.)
      AM=XM3-XM2
      BM=YM3-YM2
      CM=ZM3-ZM2
      DM23=SQRT(AM*AM+BM*BM+CM*CM)
      IF(JOP.NE.1) GOTO 6
C CHECK FOR PARALLEL LINE SOURCES
      COSTH=(XM2-XM1)*(XN2-XN1)+(YM2-YM1)*(YN2-YN1)+(ZM2-ZM1)*
     & (ZN2-ZN1)
      DM12=SQRT((XM2-XM1)**2+(YM2-YM1)**2+(ZM2-ZM1)**2)
      DN12=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
      COSTH=COSTH/(DM12*DN12)
      IF(ABS(COSTH).GE..997) GOTO 4
C     CHECK FOR PARALLEL LINE SOURCE TRANSVERSE VECTORS
      AN=XN3-XN2
      BN=YN3-YN2
      CN=ZN3-ZN2
      DN23=SQRT(AN*AN+BN*BN+CN*CN)
      COSTH=(AM*AN+BM*BN+CM*CN)/(DN23*DM23)
      IF(ABS(COSTH).GE. .997) GOTO 4
      NPT=MAX0(2*(INTP/11),2)
      GOTO 6
4     CALL PPLTS(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,
     & XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,NPT,ZMN)
      RETURN
```

```
6       D23=DM23
        NP1=NPT+1
        DX=AM/NPT
        DY=BM/NPT
        DZ=CM/NPT
        DH=D23/NPT
        DO 10 I=1,NP1
        W=3+(-1)**I
        IF(I.EQ.1.OR.I.EQ.NP1)W=W/2.
        X1=XM1+DX*(I-1)
        Y1=YM1+DY*(I-1)
        Z1=ZM1+DZ*(I-1)
        X2=XM2+DX*(I-1)
        Y2=YM2+DY*(I-1)
        Z2=ZM2+DZ*(I-1)
        IF(JOP.EQ.1)CALL ZWTPE(X1,Y1,Z1,X2,Y2,Z2,IM12,XN1,YN1,ZN1,
       & XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,NINT,Z)
        IF(JOP.EQ.2)CALL ZWTDE(X1,Y1,Z1,X2,Y2,Z2,IM12,XN1,YN1,ZN1,
       & XN2,YN2.ZN2.XN3,YN3,ZN3,NINT,BN,Z)
        IF(JOP.EQ.3)CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,IM12,XN1,YN1,ZN1,
       & XN2,YN2,ZN2,IN12,Z,0)
10      ZMN=ZMN+Z*W*COS(XK*(D23/2.-(I-1)*DH))
        ZMN=ZMN*XK*DH/(6.*SIN(XK*D23/2.))
        RETURN
        END
```

APPENDIX 17

SUBROUTINE PPLTS

```
      SUBROUTINE PPLTS(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,ZM3,IM12,
     & XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,NPT,ZMN)
      COMPLEX Z,ZMN,GAM,ETA
      COMPLEX EGD,CGDS,SGDS,SGDT,DUM,ZVSD(123)
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
C PLATE CURRENT DIRECTIONS ARE ORTHOGONAL TO A COMMON LINE
C USE FAST INTERPOLATION METHOD AND LN SINGULARITY TERM
      ZMN=(0.,0.)
      D23=SQRT((XM3-XM2)**2+(YM3-YM2)**2+(ZM3-ZM2)**2)
      E23=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
      AM=XM2-XM1
      BM=YM2-YM1
      CM=ZM2-ZM1
      AT=XN2-XN1
      BT=YN2-YN1
      CT=ZN2-ZN1
      DM12=SQRT(AM*AM+BM*BM+CM*CM)
      DN12=SQRT(AT*AT+BT*BT+CT*CT)
      AM=AM/DM12
      BM=BM/DM12
      CM=CM/DM12
      AT=AT/DN12
      BT=BT/DN12
      CT=CT/DN12
33    FORMAT(' ERROR - NPK IN PPLTS GT 120')
      EGD=CEXP(GAM*DM12)
      CGDS=(EGD+1./EGD)/2.
      SGDS=(EGD-1./EGD)/2.
      EGD=CEXP(GAM*DN12)
      SGDT=(EGD-1./EGD)/2.
      DH=AMIN1(D23,E23)/NPT
      NPI=2*IFIX(.5+.5*D23/DH)
      NPJ=2*IFIX(.5+.5*E23/DH)
      DH=D23/NPI
      EH=E23/NPJ
      NPI1=NPI+1
      NPJ1=NPJ+1
      DD=(DH+EH)/2.
      DX=(XM3-XM2)/NPI
      DY=(YM3-YM2)/NPI
      DZ=(ZM3-ZM2)/NPI
      EX=(XN3-XN2)/NPJ
      EY=(YN3-YN2)/NPJ
      EZ=(ZN3-ZN2)/NPJ
C COMPUTE DMIN,DMAX
      DMIN=1000.
      DMAX=-1000.
      DO 30 I=1,NPI1
      XM=XM1+DX*(I-1)
      YM=YM1+DY*(I-1)
      ZM=ZM1+DZ*(I-1)
      DO 30 J=1,NPJ1
      XN=XN1+EX*(J-1)
      YN=YN1+EY*(J-1)
      ZN=ZN1+EZ*(J-1)
      D=DIST(XM,YM,ZM,AM,BM,CM,XN,YN,ZN,AT,BT,CT)
      IF(D.GT.DMAX) DMAX=D
      IF(D.GT.DMIN) GOTO 30
      DMIN=D
30    CONTINUE
C FILL ZVSD WITH IMPEDANCES BETWEEN DMIN DMAX
      ZN=(XN1-XM1)*AM+(YN1-YM1)*BM+(ZN1-ZM1)*CM
      ZNN=(XN2-XM1)*AM+(YN2-YM1)*BM+(ZN2-ZM1)*CM
      IF(ABS(AM*AT+BM*BT+CM*CT).LT. .995) GOTO 100
C PARALLEL FILAMENT CASE
      XN=0.
      XNN=0.
      GOTO 200
C PARALLEL TRANSVERSE VECTOR CASE
100   DN01=(XN1-XM1)**2+(YN1-YM1)**2+(ZN1-ZM1)**2
      DN02=(XN2-XM1)**2+(YN2-YM1)**2+(ZN2-ZM1)**2
      DMN=DIST(XM1,YM1,ZM1,AM,BM,CM,XN1,YN1,ZN1,AT,BT,CT)
        AM2=(XM3-XM2)/D23
        BM2=(YM3-YM2)/D23
        CM2=(ZM3-ZM2)/D23
        AM3=BM*CM2-BM2*CM
        BM3=CM*AM2-AM*CM2
        CM3=AM*BM2-BM*AM2
        XN=(XN1-XM1)*AM3+(YN1-YM1)*BM3+(ZN1-ZM1)*CM3
        XNN=(XN2-XM1)*AM3+(YN2-YM1)*BM3+(ZN2-ZM1)*CM3
200   NPK=IFIX(1.1+(DMAX-DMIN)/DD)
      IF(NPK.GT.120) WRITE(6,33)
      DO 40 K=1,NPK
      D=DMIN+(K-1)*DD
      IF(D.LT.DD) D=DMIN+DD/2.
      CALL ZGSMM(0.,0.,0.,0.,0.,DM12,XN,D,ZN,XNN,D,ZNN,Q,DM12,
     & CGDS,SGDS,DN12,SGDT,Z)
40    ZVSD(K)=IM12*IN12*Z
      IF(DMIN.GE.DD) GOTO 45
C TAKE CARE OF LOGARITHMIC SINGULARITY
      RZ=REAL(ZVSD(1))
      X1=AIMAG(ZVSD(1))
      X2=AIMAG(ZVSD(2))
      C2=(X2-X1)/ALOG((DMIN+DD)/(DMIN+DD/2.))
      C1=X1-C2*ALOG(DMIN+DD/2.)
      AIZ=2.*C1+C2*(ALOG(DD*DD+DMIN*DMIN)-2.)+2.*C2*DMIN*
     & ATAN2(DD,DMIN)/DD-(C1+C2*ALOG(SQRT(DMIN*DMIN+DD*DD)))
      ZVSD(1)=CMPLX(RZ,AIZ)
```

```
45    ZVSD(NPK+1)=ZVSD(NPK)
C DO 2-D SIMPSON INTEGRATION
      DO 90 I=1,NPI1
      W=3+(-1)**I
      IF(I.EQ.1 .OR. I.EQ.NPI1) W=W/2.
      XM=XM1+DX*(I-1)
      YM=YM1+DY*(I-1)
      ZM=ZM1+DZ*(I-1)
      DO 90 J=1,NPJ1
      V=3+(-1)**J
      IF(J.EQ.1 .OR. J.EQ.NPJ1) V=V/2.
      XN=XN1+EX*(J-1)
      YN=YN1+EY*(J-1)
      ZN=ZN1+EZ*(J-1)
C COMPUTE DISTANCE BETWEEN THE TWO MONOPOLES
      D=DIST(XM,YM,ZM,AM,BM,CM,XN,YN,ZN,AT,BT,CT)
      N=ABS(D-DMIN)/DD+1
      Z=ZVSD(N)+(ZVSD(N+1)-ZVSD(N))/DD*(D-DMIN-(N-1)*DD)
      Z=Z*W*COS(XK*(D23/2.-(I-1)*DH))*XK*DH/(6.*SIN(XK*D23/2.))
      Z=Z*V*COS(XK*(E23/2.-(J-1)*EH))*XK*EH/(6.*SIN(XK*E23/2.))
90    ZMN=ZMN+Z
      RETURN
      END
      FUNCTION DIST(X1,Y1,Z1,A1,B1,C1,X2,Y2,Z2,A2,B2,C2)
C DISTANCE BETWEEN LINE IN DIRECTION(A1,B1,C1) THROUGH
C P1 TO LINE IN DIRECTION (A2,B2,C2) THROUGH P2
      A3=B1*C2-B2*C1
      B3=C1*A2-A1*C2
      C3=A1*B2-B1*A2
      T1=(Y2-Y1)*C1-(Z2-Z1)*B1
      T2=(Z2-Z1)*A1-(X2-X1)*C1
      T3=(X2-X1)*B1-(Y2-Y1)*A1
      D=A3*A3+B3*B3+C3*C3
      IF(D.GT.1.E-6) GOTO 10
C PARALLEL LINES
      DIST=SQRT(T1*T1+T2*T2+T3*T3)
      RETURN
C NOT PARALLEL
10    DIST=ABS(A2*T1+B2*T2+C2*T3)/SQRT(D)
      RETURN
      END
```

APPENDIX 18

SUBROUTINE PLTST2

```
         SUBROUTINEPLTST2(XM1,YM1,ZM1,XM2,YM2,ZM2,XM3,YM3,
      2  ZM3,XM4,YM4,ZM4,IM12,JOP,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,
      3  YN3,ZN3,XN4,YN4,ZN4,IN12,NPT,NINT,IACM,
      &  IACN,ZMN)
C    THIS ROUTINE ONLY WORKS WHEN DISTANCES BETWEEN POINTS 1 AND 2
C    ARE GREATER THAN ZERO.
         DIMENSION SX(125),SY(125),SY1(125),SY2(125),SY3(125),SE(125)
         DIMENSION SW(125)
         DIMENSION SYR(125)
         COMPLEX ZVSD1,ZVSD2
         COMPLEX Z,ZMN,GAM,ETA,RIT,EGD
         COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
         IWZ=0
         ISDTCH=0
         JSDTCH=0
         MPT=NPT
         MINT=NINT
         KINT=0
         IF(NPT.GT.1)GO TO 15
         X1=(XM1+XM2)/2.0
         Y1=(YM1+YM2)/2.0
         Z1=(ZM1+ZM2)/2.0
         X2=(XM3+XM4)/2.0
         Y2=(YM3+YM4)/2.0
         Z2=(ZM3+ZM4)/2.0
         D12=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
         GO TO 79
  15     CONTINUE
         ZMN=(0.,0.)
         DM43=SQRT((XM3-XM4)**2+(YM3-YM4)**2+(ZM3-ZM4)**2)
         DM12=SQRT((XM2-XM1)**2+(YM2-YM1)**2+(ZM2-ZM1)**2)
         DM23=SQRT((XM3-XM2)**2+(YM3-YM2)**2+(ZM3-ZM2)**2)
         DM14=SQRT((XM4-XM1)**2+(YM4-YM1)**2+(ZM4-ZM1)**2)
         DMMX=AMAX1(DM14,DM23)
  62     CONTINUE
         NP1=MPT+1
         DXT=(XM2-XM1)/MPT
         DYT=(YM2-YM1)/MPT
         DZT=(ZM2-ZM1)/MPT
         DXE=(XM3-XM4)/MPT
```

```
        DYE=(YM3-YM4)/MPT
        DZE=(ZM3-ZM4)/MPT
        DHT=DM12/MPT
        FM=0.
        DO 10 IDO=1,NP1
        I=IDO
        WT=3.0+(-1)**I
        IF(I.EQ.1.OR.I.EQ.NP1)WT=WT/2.0
        SX(I)=(I-1)*DHT
        X1=XM1+DXT*(I-1)
        Y1=YM1+DYT*(I-1)
        Z1=ZM1+DZT*(I-1)
        X2=XM4+DXE*(I-1)
        Y2=YM4+DYE*(I-1)
        Z2=ZM4+DZE*(I-1)
        D12=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
        IF(D12.GT.Q)GO TO 49
        Z=(.0,.0)
        GO TO 200
  49    CONTINUE
79      CONTINUE
        IF(JOP.EQ.1)CALL ZWTPE2(X1,Y1,Z1,X2,Y2,Z2,D12,1,
     1  IM12,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,
     2  XN4,YN4,ZN4,IN12,MINT,IACN,Z,KINT)
        IF(NPT.GT.1.OR.JOP.NE.1)GO TO 104
        XNC=(XN1+XN2+XN3+XN4)/4.0
        YNC=(YN1+YN2+YN3+YN4)/4.0
        ZNC=(ZN1+ZN2+ZN3+ZN4)/4.0
        DM1NC=SQRT((XNC-XM1)**2+(YNC-YM1)**2+(ZNC-ZM1)**2)
        DM4NC=SQRT((XNC-XM4)**2+(YNC-YM4)**2+(ZNC-ZM4)**2)
        D1NC=SQRT((XNC-X1)**2+(YNC-Y1)**2+(ZNC-Z1)**2)
        D2NC=SQRT((XNC-X2)**2+(YNC-Y2)**2+(ZNC-Z2)**2)
        DM2NC=SQRT((XNC-XM2)**2+(YNC-YM2)**2+(ZNC-ZM2)**2)
        DM3NC=SQRT((XNC-XM3)**2+(YNC-YM3)**2+(ZNC-ZM3)**2)
        DAV1=0.5*(DM1NC+DM4NC)
        DAVG=0.5*(D1NC+D2NC)
        DAV2=0.5*(DM2NC+DM3NC)
        EGD=1.0/(CEXP(GAM*DAVG)*DAVG)
        RIT=1.0/(CEXP(GAM*DAV1)*DAV1)+4.0*EGD+1.0/(CEXP(GAM*DAV2)*DAV2)
        ZMN=Z*RIT/(6.0*EGD)
        RETURN
 104    CONTINUE
        IF(JOP.NE.3)GO TO 206
        DS=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
        CAS=(X2-X1)/DS
        CBS=(Y2-Y1)/DS
        CGS=(Z2-Z1)/DS
        SZ1=(XN1-X1)*CAS+(YN1-Y1)*CBS+(ZN1-Z1)*CGS
        SZ2=(XN2-X1)*CAS+(YN2-Y1)*CBS+(ZN2-Z1)*CGS
        RHO1=SQRT((XN1-X1-SZ1*CAS)**2+(YN1-Y1-SZ1*CBS)**2
     & +(ZN1-Z1-SZ1*CGS)**2)
        RHO2=SQRT((XN2-X1-SZ2*CAS)**2+(YN2-Y1-SZ2*CBS)**2
     & +(ZN2-Z1-SZ2*CGS)**2)
        RHOM=AMIN1(RHO1,RHO2)
        SIGN=1.0
        IF(RHOM.GT.0.1*DHT)GO TO 160
        XCL=XN1
        YCL=YN2
```

```
      ZCL=ZN2
      IF(RHO1.LT.RHO2)GO TO 105
      XCL=XN2
      YCL=YN2
      ZCL=ZN2
 105  CONTINUE
      DSCL1=SQRT((XCL-X1)**2+(YCL-Y1)**2+(ZCL-Z1)**2)
      DSCL2=SQRT((XCL-X2)**2+(YCL-Y2)**2+(ZCL-Z2)**2)
      IF(DSCL1.GT.DS+RHOM.OR.DSCL2.GT.DS+RHOM)GO TO 160
      IF(I.GT.1)GO TO 110
      X1=XM1+DXT
      Y1=YM1+DYT
      Z1=ZM1+DZT
      X2=XM4+DXE
      Y2=YM4+DYE
      Z2=ZM4+DZE
      CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,IM12,XN1,YN1,ZN1,
     & XN2,YN2,ZN2,IN12,ZVSD2,0)
      SIGN=-1.0
 110  CONTINUE
      X1=XM1+(I-1)*DXT+SIGN*0.5*DXT
      Y1=YM1+(I-1)*DYT+SIGN*0.5*DYT
      Z1=ZM1+(I-1)*DZT+SIGN*0.5*DZT
      X2=XM4+(I-1)*DXE+SIGN*0.5*DXE
      Y2=YM4+(I-1)*DYE+SIGN*0.5*DYE
      Z2=ZM4+(I-1)*DZE+SIGN*0.5*DZE
      CALL ZWTWE(X1,Y1,Z1,X2.Y2,Z2,IM12,XN1,YN1,ZN1,
     & XN2,YN2,ZN2,IN12,ZVSD1,0)
      RZ=REAL(ZVSD1)
      XZ1=AIMAG(ZVSD1)
      XZ2=AIMAG(ZVSD2)
      C2=(XZ2-XZ1)/ALOG(2.0)
      C1=XZ1-C2*ALOG(DHT/2.)
      AIZ=2.*C1+2.*C2*(ALOG(DHT)-1.)-XZ2
      SY(I)=AIZ
      SYR(I)=RZ
      GO TO 10
 160  CONTINUE
      CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,IM12,XN1,YN1,ZN1,
     2 XN2,YN2,ZN2,IN12,Z.0)
      ZVSD2=Z
 200  CONTINUE
      SY(I)=AIMAG(Z)
      SYR(I)=REAL(Z)
206   ZMN=ZMN+Z*WT
      FM=FM+WT
  10  CONTINUE
      ZMN=ZMN/FM
      IF(JOP.NE.3)RETURN
      ZVSR=SPLINT(SX,SYR,NP1,.0,DM12,SY1,SY2,SY3,SE,IND)
     & /DM12
      ZVS=SPLINT(SX,SY,NP1,.0,DM12,SY1,SY2,SY3,SE,IND)
     & /DM12
      ZMN=CMPLX(ZVSR,ZVS)
      RETURN
      END
```

APPENDIX 19

SUBROUTINE ZWTPE

```
      SUBROUTINE ZWTPE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,
     & XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,INTP,ZMN)
      COMPLEX CGDM,SGDM,SGDN,EGD,ETA,GAM,ZMN,P11,DUM
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
      DM=SQRT((XM1-XM2)**2+(YM1-YM2)**2+(ZM1-ZM2)**2)
      EGD=CEXP(GAM*DM)
      CGDM=(EGD+1./EGD)/2.
      SGDM=(EGD-1./EGD)/2.
      WN=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
      DN=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
      EGD=CEXP(GAM*DN)
      SGDN=(EGD-1./EGD)/2.
C CHECK DIST BETWEEN TEST & EXP
      D=SQRT(((XM1+XM2-XN3-XN1)**2+(YM1+YM2-YN3-YN1)**2
     & +(ZM1+ZM2-ZN3-ZN1)**2)/4.)
      DD=(DM+SQRT(WN*WN+DN*DN))/1.8
      NPLS=INTP
      IF(D.GT.DD) NPLS=MAX0(2*(INTP/11),2)
      DX=(XN3-XN2)/NPLS
      DY=(YN3-YN2)/NPLS
      DZ=(ZN3-ZN2)/NPLS
      NP1=NPLS+1
      ZMN=(0.,0.)
      DO 10 I=1,NP1
      W=3+(-1)**I
      IF(I.EQ.1 .OR. I.EQ.NP1) W=W/2.
      X1=XN1+(I-1)*DX
      Y1=YN1+(I-1)*DY
      Z1=ZN1+(I-1)*DZ
      X2=XN2+(I-1)*DX
      Y2=YN2+(I-1)*DY
      Z2=ZN2+(I-1)*DZ
     &CALL ZGSMM(XM1,YM1,ZM1,XM2,YM2,ZM2,X1,Y1,Z1,X2,Y2,Z2,Q,
     & DM,CGDM,SGDM,DN,SGDN,P11)
      ZMN=ZMN+W*P11*COS(XK*WN*(.5-(I-1.)/NPLS))
10    CONTINUE
      ZMN=ZMN*WN/(3.*NPLS)*IN12*IM12*XK/(2.*SIN(XK*WN/2.))
      RETURN
      END
```

APPENDIX 20

SUBROUTINE ZWTDE

```
      SUBROUTINE ZWTDE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,
     & XN0,YN0,ZN0.XN1.YN1,ZN1,XN2,YN2,ZN2,INTD,B,ZMN)
      COMPLEX ZMN,P11,DUM,EGD,ETA,GAM,SGDM,CGDM,SGDN
      REAL L1,L2,L3,M1,M2,M3,N1,N2,N3
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
C DEFINE PRIME COORDS ON PLANE OF DISK
      DNR=SQRT((XN2-XN0)**2+(YN2-YN0)**2+(ZN2-ZN0)**2)
      L1=(XN2-XN0)/DNR
      M1=(YN2-YN0)/DNR
      N1=(ZN2-ZN0)/DNR
      L3=(YN1-YN0)*(ZN2-ZN0)-(YN2-YN0)*(ZN1-ZN0)
      M3=(ZN1-ZN0)*(XN2-XN0)-(ZN2-ZN0)*(XN1-XN0)
      N3=(XN1-XN0)*(YN2-YN0)-(XN2-XN0)*(YN1-YN0)
      RN=SQRT(L3*L3+M3*M3+N3*N3)
      L3=L3/RN
      M3=M3/RN
      N3=N3/RN
      L2=M3*N1-N3*M1
      M2=L1*N3-N1*L3
      N2=L3*M1-L1*M3
C DEFINE PARAMETERS FOR GGS
      DM=SQRT((XM1-XM2)**2+(YM1-YM2)**2+(ZM1-ZM2)**2)
      DN=B-A
      EGD=CEXP(GAM*DM)
      SGDM=(EGD-1./EGD)/2.
      CGDM=(EGD+1./EGD)/2.
      EGD=CEXP(GAM*DN)
      SGDN=(EGD-1./EGD)/2.
C CHECK DIST BETWEEN TEST AND EXP
      D=SQRT(((XM1+XM2)/2.-XN0)**2+((YM1+YM2)/2.-YN0)**2
     & +((ZM1+ZM2)/2.-ZN0)**2)
      DD=B+DM/1.8
      NDLS=INTD
      IF(D.GT.DD) NDLS=2*(INTD/6)
      IF(NDLS.LT.2)NDLS=2
      NP1=NDLS+1
      DPH=2.*PI/NDLS
      ZMN=(0.,0.)
      DO 10 I=1,NP1
      W=3+(-1)**I
      IF(I.EQ.1 .OR. I.EQ.NP1) W=W/2.
      PH=(I-1)*DPH
      XP1=A*COS(PH)
      YP1=A*SIN(PH)
      XP2=B*COS(PH)
      YP2=B*SIN(PH)
C TRANSFORM COORDS TO ORIGINAL SYSTEM
      X1=L1*XP1+L2*YP1+XN0
      Y1=M1*XP1+M2*YP1+YN0
      Z1=N1*XP1+N2*YP1+ZN0
      X2=L1*XP2+L2*YP2+XN0
      Y2=M1*XP2+M2*YP2+YN0
      Z2=N1*XP2+N2*YP2+ZN0
      CALL ZGSMM(XM1,YM1,ZM1,XM2,YM2,ZM2,X1,Y1,Z1,X2,Y2,Z2,
     & Q,DM,CGDM,SGDM,DN,SGDN,P11)
      ZMN=ZMN-W*P11
10    CONTINUE
      ZMN=ZMN*IM12*DPH/(6.*PI)
      RETURN
      END
```

APPENDIX 21

SUBROUTINE ZWTWE

```
 SUBROUTINE ZWTWE(XM1,YM1,ZM1,XM2,YM2,ZM2,IM12,XN1,YN1,ZN1,
&XN2,YN2,ZN2,IN12,ZMN,IWW)
 COMPLEX ZMN,P11,DUM,EGD,SGDM,CGDM,SGDN,ETA,GAM
 COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
 DM=SQRT((XM1-XM2)**2+(YM1-YM2)**2+(ZM1-ZM2)**2)
 DN=SQRT((XN1-XN2)**2+(YN1-YN2)**2+(ZN1-ZN2)**2)
 EGD=CEXP(GAM*DM)
 SGDM=(EGD-1./EGD)/2.
 CGDM=(EGD+1./EGD)/2.
 EGD=CEXP(GAM*DN)
 SGDN=(EGD-1./EGD)/2.
 AA=Q
 IF(IWW.EQ.1) AA=A
 CALL ZGSMM(XM1,YM1,ZM1,XM2,YM2,ZM2,XN1,YN1,ZN1,XN2,YN2,ZN2,
&AA,DM,CGDM,SGDM,DN,SGDN,P11)
 ZMN=P11*IN12*IM12
 RETURN
 END
```

APPENDIX 22

SUBROUTINE ZWTPE2

```
      SUBROUTINE ZWTPE2(XM1,YM1,ZM1,XM2,YM2,ZM2,DM,
     2 IM12,XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,
     3 XN4,YN4,ZN4,IN12,NPLS,IACN,ZMN,KINT)
      DIMENSION SX(125),SY(125),SY1(125),SY2(125),SY3(125)
      DIMENSION SW(125),SE(125),SYR(125)
      COMPLEX ZMN,P11,DUM,EGD,ETA,GAM,SGDM,CGDM,SGDN,ZP11
      COMPLEX ZVSD1,ZVSD2,ZVSD3,RIT,YP11
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
      CAM=(XM2-XM1)/DM
      CBM=(YM2-YM1)/DM
      CGM=(ZM2-ZM1)/DM
      CDKS=COS(XK*DM)
      SDKS=SIN(XK*DM)
      WNT=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
      WNE=SQRT((XN3-XN4)**2+(YN3-YN4)**2+(ZN3-ZN4)**2)
      CAST=(XN2-XN1)/WNT
      CBST=(YN2-YN1)/WNT
      CGST=(ZN2-ZN1)/WNT
      CASE=(XN3-XN4)/WNE
      CBSE=(YN3-YN4)/WNE
      CGSE=(ZN3-ZN4)/WNE
      IF(NPLS.NE.1)GO TO 31
      NP1=1
      HHT=WNT/2.0
      HHE=WNE/2.0
      HH=HHT
      ILS=1
      ISM=0
      I11=0
      GO TO 119
   31 CONTINUE
      DN14=SQRT((XN4-XN1)**2+(YN4-YN1)**2+(ZN4-ZN1)**2)
      DN23=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
      DNMX=AMAX1(DN14,DN23)
      HHT=WNT/NPLS
      HHE=WNE/NPLS
      HH=HHT
      NP1=NPLS+1
      DD=HHT
      SLIMT=WNT
      IF(NP1.GT.90)WRITE(11,16)NP1
 16   FORMAT(//5X,'***WARNING NP1 IN SUB. ZWTPE IS TOO LARGE:',
     & '   NPT=',I5,'***'//)
      ZMN=(.0,.0)
      I11=0
      DO 12 I=1,NP1
      ILS=I-1
      GGT=HHT
      GGE=HHE
      SX(I)=ILS*HH
      IF(I11.GT.0)GO TO 8
119   X1=XN1+(ILS*HHT)*CAST
      Y1=YN1+(ILS*HHT)*CBST
      Z1=ZN1+(ILS*HHT)*CGST
      X2=XN4+(ILS*HHE)*CASE
      Y2=YN4+(ILS*HHE)*CBSE
      Z2=ZN4+(ILS*HHE)*CGSE
      DN12=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
      DN120=DN12
      IF(DN12.GT.Q)GO TO 215
      P11=(.0,.0)
      GO TO 8
215   CONTINUE
      DDMN=0.0
      IF(NP1.NE.1)GO TO 22
      DMCNC1=SQRT((0.5*(XM1+XM2-XN1-XN4))**2+(0.5*(YM1+YM2-YN1-YN4))
     & **2+(0.5*(ZM1+ZM2-ZN1-ZN4))**2)
      DMCNC2=SQRT((0.5*(XM1+XM2-XN2-XN3))**2+(0.5*(YM1+YM2-YN2-YN3))
     & **2+(0.5*(ZM1+ZM2-ZN2-ZN3))**2)
      IF(DMCNC1.GT.DMCNC2)GO TO 25
      DMCNC=DMCNC1
      DM1N1=SQRT((XM1-XN1)**2+(YM1-YN1)**2+(ZM1-ZN1)**2)
      DM1N2=SQRT((XM1-XN4)**2+(YM1-YN4)**2+(ZM1-ZN4)**2)
      DM2N1=SQRT((XM2-XN1)**2+(YM2-YN1)**2+(ZM2-ZN1)**2)
      DM2N2=SQRT((XM2-XN4)**2+(YM2-YN4)**2+(ZM2-ZN4)**2)
      GO TO 26
 25   CONTINUE
      DMCNC=DMCNC2
      DM1N1=SQRT((XM1-XN2)**2+(YM1-YN2)**2+(ZM1-ZN2)**2)
      DM1N2=SQRT((XM1-XN3)**2+(YM1-YN3)**2+(ZM1-ZN3)**2)
      DM2N1=SQRT((XM2-XN2)**2+(YM2-YN2)**2+(ZM2-ZN2)**2)
      DM2N2=SQRT((XM2-XN3)**2+(YM2-YN3)**2+(ZM2-ZN3)**2)
 26   DDMN=AMIN1(DMCNC,DM1N1,DM1N2,DM2N1,DM2N2)
      DM1NC=SQRT((XM1-0.5*(X1+X2))**2+(YM1-0.5*(Y1+Y2))
     & **2+(ZM1-0.5*(Z1+Z2))**2)
      DM2NC=SQRT((XM2-0.5*(X1+X2))**2+(YM2-0.5*(Y1+Y2))
     & **2+(ZM2-0.5*(Z1+Z2))**2)
      DAVG=0.5*(DM1NC+DM2NC)
      DM1NA=SQRT((XM1-0.5*(XN1+XN4))**2+(YM1-0.5*(YN1+YN4))
     & **2+(ZM1-0.5*(ZN1+ZN4))**2)
      DM2NA=SQRT((XM2-0.5*(XN1+XN4))**2+(YM2-0.5*(YN1+YN4))
     & **2+(ZM2-0.5*(ZN1+ZN4))**2)
      DAV1=0.5*(DM1NA+DM2NA)
      DM1NB=SQRT((XM1-0.5*(XN2+XN3))**2+(YM1-0.5*(YN2+YN3))
     & **2+(ZM1-0.5*(ZN2+ZN3))**2)
      DM2NB=SQRT((XM2-0.5*(XN2+XN3))**2+(YM2-0.5*(YN2+YN3))
     & **2+(ZM2-0.5*(ZN2+ZN3))**2)
      DAV2=0.5*(DM1NB+DM2NB)
```

```
        GO TO 87
  22    CONTINUE
        DM1N1=SQRT((XM1-X1)**2+(YM1-Y1)**2+(ZM1-Z1)**2)
        DM1N2=SQRT((XM1-X2)**2+(YM1-Y2)**2+(ZM1-Z2)**2)
        DM2N1=SQRT((XM2-X1)**2+(YM2-Y1)**2+(ZM2-Z1)**2)
        DM2N2=SQRT((XM2-X2)**2+(YM2-Y2)**2+(ZM2-Z2)**2)
        DMCNC=SQRT((0.5*(XM2+XM1-X2-X1))**2+(0.5*(YM2+YM1-Y2-Y1))**2+
     & (0.5*(ZM2+ZM1-Z2-Z1))**2)
        DDMN=AMIN1(DM1N1,DM1N2,DM2N1,DM2N2)
        CAN=(X2-X1)/DN12
        CBN=(Y2-Y1)/DN12
        CGN=(Z2-Z1)/DN12
        DPM1=ABS(CAM*CAN+CBM*CBN+CGM*CGN)
  85    CONTINUE
C   CHECK DIST BETWEEN TEST AND EXP
        INT=0
        IF(KINT.EQ.1)GO TO 86
        IF(DMCNC.GT.0.5*(DM+DN12)+0.15*WV)GO TO 89
        SZDA=(X1-XM1)*CAM+(Y1-YM1)*CBM+(Z1-ZM1)*CGM
        XZDA=X1-XM1-SZDA*CAM
        YZDA=Y1-YM1-SZDA*CBM
        ZZDA=Z1-ZM1-SZDA*CGM
        RHODA=SQRT(XZDA*XZDA+YZDA*YZDA+ZZDA*ZZDA)
        SZDB=(X2-XM1)*CAM+(Y2-YM1)*CBM+(Z2-ZM1)*CGM
        SZC1=(XM1-X1)*CAN+(YM1-Y1)*CBN+(ZM1-Z1)*CGN
        SZC2=(XM2-X1)*CAN+(YM2-Y1)*CBN+(ZM2-Z1)*CGN
        IF(DPM1.LT.0.999)GO TO 156
        IF(DN12.GT.DM)GO TO 145
        IF(SZDA.GT.0.0.AND.SZDA.LT.DM)GO TO 155
        IF(SZDB.GT.0.0.AND.SZDB.LT.DM)GO TO 155
        GO TO 156
 145    IF(SZC1.GT.0.0.AND.SZC1.LT.DN12)GO TO 155
        IF(SZC2.GT.0.0.AND.SZC2.LT.DN12)GO TO 155
        GO TO 156
 155    IF(RHODA.LT.0.12*WV)GO TO 86
        IF(RHODA.LT.0.20*WV)GO TO 87
        GO TO 89
 156    CONTINUE
        XZDB=X2-XM1-SZDB*CAM
        YZDB=Y2-YM1-SZDB*CBM
        ZZDB=Z2-ZM1-SZDB*CGM
        RHODB=SQRT(XZDB*XZDB+YZDB*YZDB+ZZDB*ZZDB)
        XZC1=XM1-X1-SZC1*CAN
        YZC1=YM1-Y1-SZC1*CBN
        ZZC1=ZM1-Z1-SZC1*CGN
        RHOC1=SQRT(XZC1*XZC1+YZC1*YZC1+ZZC1*ZZC1)
        XZC2=XM2-X1-SZC2*CAN
        YZC2=YM2-Y1-SZC2*CBN
        ZZC2=ZM2-Z1-SZC2*CGN
        RHOC2=SQRT(XZC2*XZC2+YZC2*YZC2+ZZC2*ZZC2)
        DTDADB=XZDA*XZDB+YZDA*YZDB+ZZDA*ZZDB
        DTC1C2=XZC1*XZC2+YZC1*YZC2+ZZC1*ZZC2
        IF(DTDADB.GT.0.0.OR.DTC1C2.GT.0.0)GO TO 105
        RHOMIN=AMIN1(RHODA,RHODB,RHOC1,RHOC2)
        IF(RHOMIN.LE.0.05*WV)GO TO 86
        CAP=CBN*CGM-CGN*CBM
        CBP=CGN*CAM-CAN*CGM
        CGP=CAN*CBM-CBN*CAM
```

```
      DCP=SQRT(CAP*CAP+CBP*CBP+CGP*CGP)
      DPP=ABS((XM1-X1)*CAP+(YM1-Y1)*CBP+(ZM1-Z1)*CGP)/DCP
      IF(DPP.LE.0.05*WV)GO TO 86
      IF(DPP.LE.0.1*WV)GO TO 87
      INT=2
      GO TO 86
105   CONTINUE
      IF(DTDADB.GT.0.0)GO TO 107
      RHONP=AMIN1(RHOC1,RHOC2)
      GO TO 109
107   IF(DTC1C2.GT.0.0)GO TO 108
      RHONP=AMIN1(RHODA,RHODB)
      GO TO 109
108   RHONP=DDMN
109   CONTINUE
      IF(RHONP.LE.0.05*WV)GO TO 86
      IF(RHONP.LE.0.1*WV)GO TO 87
 89   INT=2
      GO TO 86
 87   INT=4
 86   CONTINUE
      SDKT=SIN(XK*DN12)
      SZM1=(X1-XM1)*CAM+(Y1-YM1)*CBM+(Z1-ZM1)*CGM
      CALL GGS1(XM1,YM1,ZM1,XM2,YM2,ZM2,X1,Y1,Z1,X2,Y2,Z2,XK,
     2DM,CDKS,SDKS,DN12,SDKT,INT,ETA,GAM,P11,DUM,DUM,DUM)
      IF(NP1.GT.1)GO TO 97
      EGD=1.0/(CEXP(GAM*DAVG)*DAVG)
      RIT=1.0/(CEXP(GAM*DAV1)*DAV1)+4.0*EGD+1.0/(CEXP(GAM*DAV2)*DAV2)
      ZMN=P11*IM12*IN12*RIT/(6.0*EGD)
      RETURN
 97   CONTINUE
      ZVSD2=P11
  8   CONTINUE
      SYR(I)=REAL(P11)
      SY(I)=AIMAG(P11)
      I11=0
 12   CONTINUE
      ZVSR=SPLINT(SX,SYR,NP1,.0,SLIMT,SY1,SY2,SY3,SE,IND)
     & /SLIMT
      ZVST=SPLINT(SX,SY,NP1,.0,SLIMT,SY1,SY2,SY3,SE,IND)
     & /(SLIMT)
      ZMN=IM12*IN12*CMPLX(ZVSR,ZVST)
      RETURN
      END
```

APPENDIX 23

SUBROUTINE DSKTST

```
      SUBROUTINEDSKTST(XM0,YM0,ZM0,XM1,YM1,ZM1,XM2,YM2,ZM2,JOP,
     &XN1,YN1,ZN1,XN2.YN2.ZN2,XN3,YN3,ZN3,IN12,NINT,INTD,
     &BM,BN.ZMN)
      COMPLEX Z,ZMN,GAM,ETA
      REAL L1,L2,L3,M1,M2,M3,N1,N2,N3
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
      DNR=SQRT((XM2-XM0)**2+(YM2-YM0)**2+(ZM2-ZM0)**2)
      L1=(XM2-XM0)/DNR
      M1=(YM2-YM0)/DNR
      N1=(ZM2-ZM0)/DNR
      L3=(YM1-YM0)*(ZM2-ZM0)-(YM2-YM0)*(ZM1-ZM0)
      M3=(ZM1-ZM0)*(XM2-XM0)-(ZM2-ZM0)*(XM1-XM0)
      N3=(XM1-XM0)*(YM2-YM0)-(XM2-XM0)*(YM1-YM0)
      RN=SQRT(L3*L3+M3*M3+N3*N3)
      L3=L3/RN
      M3=M3/RN
      N3=N3/RN
      L2=M3*N1-N3*M1
      M2=L1*N3-N1*L3
      N2=L3*M1-L1*M3
C CHECK DISTANCE BETWEEN TEST AND EXP
      IF(JOP.NE.3)GO TO 100
      XN3=XN2
      YN3=YN2
      ZN3=ZN2
100   CONTINUE
      D=SQRT(((XN1+XN3)/2.-XM0)**2+((YN1+YN3)/2.-YM0)**2
     & +((ZN1+ZN3)/2.-ZM0)**2)
      NDT=INTD
      IF(D.GT. .5*WV) NDT=2*(INTD/6)
      IF(NDT.LT.2)NDT=2
      DPH=2.*PI/NDT
      ZMN=(0.,0.)
      DO 10 I=1,NDT
      W=3+(-1)**I
      PH=(I-1)*DPH
      XP1=A*COS(PH)
      YP1=A*SIN(PH)
      XP2=BM*COS(PH)
      YP2=BM*SIN(PH)
```

```
C TRANSFORM COORDS TO ORIGINAL SYSTEM
      X1=L1*XP1+L2*YP1+XM0
      Y1=M1*XP1+M2*YP1+YM0
      Z1=N1*XP1+N2*YP1+ZM0
      X2=L1*XP2+L2*YP2+XM0
      Y2=M1*XP2+M2*YP2+YM0
      Z2=N1*XP2+N2*YP2+ZM0
      IF(JOP.EQ.1)CALL ZWTPE(X1,Y1,Z1,X2,Y2,Z2,-1,XN1,YN1,ZN1,
     & XN2,YN2,ZN2,XN3,YN3,ZN3,IN12,NINT,Z)
      IF(JOP.EQ.2)CALL ZWTDE(X1,Y1,Z1,X2,Y2,Z2,-1,XN1,YN1,ZN1,
     & XN2,YN2,ZN2,XN3,YN3,ZN3,NINT,BN,Z)
      IF(JOP.EQ.3)CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,-1,XN1,YN1,ZN1,
     & XN2,YN2,ZN2,IN12,Z,0)
      ZMN=ZMN+Z*W
10    CONTINUE
      ZMN=ZMN/(3.*NDT)
      RETURN
      END
```

APPENDIX 24

SUBROUTINE DSKTS2

```
      SUBROUTINEDSKTS2(XM0,YM0,ZM0,XM1,YM1,ZM1,XM2,YM2,ZM2,JOP,
     & XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,YN4,ZN4,IN12,NINT,
     & NDT,BM,BN,ZMN)
      COMPLEX Z,ZMN.GAM,ETA
      REAL L1,L2,L3,M1,M2,M3,N1,N2,N3
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
      DNR=SQRT((XM2-XM0)**2+(YM2-YM0)**2+(ZM2-ZM0)**2)
      L1=(XM2-XM0)/DNR
      M1=(YM2-YM0)/DNR
      N1=(ZM2-ZM0)/DNR
      L3=(YM1-YM0)*(ZM2-ZM0)-(YM2-YM0)*(ZM1-ZM0)
      M3=(ZM1-ZM0)*(XM2-XM0)-(ZM2-ZM0)*(XM1-XM0)
      N3=(XM1-XM0)*(YM2-YM0)-(XM2-XM0)*(YM1-YM0)
      RN=SQRT(L3*L3+M3*M3+N3*N3)
      L3=L3/RN
      M3=M3/RN
      N3=N3/RN
      L2=M3*N1-N3*M1
      M2=L1*N3-N1*L3
      N2=L3*M1-L1*M3
      DPH=2.*PI/NDT
      IFGD=0
      IPRL=0
      DK=0.0
      IQQ=0
      IF(JOP.NE.1)GO TO 5
      DMR1=SQRT((XM1-XM0)**2+(YM1-YM0)**2+(ZM1-ZM0)**2)
      DMR2=SQRT((XM2-XM0)**2+(YM2-YM0)**2+(ZM2-ZM0)**2)
      STM=SQRT(((YM1-YM0)*(ZM2-ZM0)-(YM2-YM0)*(ZM1-ZM0))**2+
     2((XM2-XM0)*(ZM1-ZM0)-(XM1-XM0)*(ZM2-ZM0))**2+
     3((XM1-XM0)*(YM2-YM0)-(XM2-XM0)*(YM1-YM0))**2)/(DMR1*DMR2)
      DN12=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
      DN23=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
      STN=SQRT(((YN2-YN1)*(ZN3-ZN2)-(YN3-YN2)*(ZN2-ZN1))**2+
     2((XN3-XN2)*(ZN2-ZN1)-(XN2-XN1)*(ZN3-ZN2))**2+
     3((XN2-XN1)*(YN3-YN2)-(XN3-XN2)*(YN2-YN1))**2)/(DN12*DN23)
      CAS=(((YN2-YN1)*(ZN3-ZN2)-(YN3-YN2)*(ZN2-ZN1))*((YM1-YM0)*
     2(ZM2-ZM0)-(YM2-YM0)*(ZM1-ZM0)))/(DMR1*DMR2*DN12*DN23*STN*STM)
      CBS=(((XN3-XN2)*(ZN2-ZN1)-(XN2-XN1)*(ZN3-ZN2))*((XM2-XM0)*
     2(ZM1-ZM0)-(XM1-XM0)*(ZM2-ZM0)))/(DMR1*DMR2*DN12*DN23*STN*STM)
```

```
        CGS=(((XN2-XN1)*(YN3-YN2)-(XN3-XN2)*(YN2-YN1))*((XM1-XM0)*
     2(YM2-YM0)-(XM2-XM0)*(YM1-YM0)))/(DMR1*DMR2*DN12*DN23*STN*STM)
        IF(ABS(CAS+CBS+CGS).LT.0.999)GO TO 5
        DK=(XM1-XN1)*CAS+(YM1-YN1)*CBS+(ZM1-ZN1)*CGS
        IF(ABS(DK).LT.Q)IQQ=1
 5      CONTINUE
        ZMN=(.0,.0)
  14    CONTINUE
        DO 10 IDO=1,NDT
        I=IDO
  18    CONTINUE
        W=3.+(-1)**I
        PH=(I-1)*DPH
        XP1=A*COS(PH)
        YP1=A*SIN(PH)
        XP2=BM*COS(PH)
        YP2=BM*SIN(PH)
C TRANSFORM COORDS TO ORIGINAL SYSTEM
        IF(IQQ.EQ.0)GO TO 8
        X1=L1*XP1+L2*YP1+L3*Q+XM0
        Y1=M1*XP1+M2*YP1+M3*Q+YM0
        Z1=N1*XP1+N2*YP1+N3*Q+ZM0
        X2=L1*XP2+L2*YP2+L3*Q+XM0
        Y2=M1*XP2+M2*YP2+M3*Q+YM0
        Z2=N1*XP2+N2*YP2+N3*Q+ZM0
        GO TO 9
 8      CONTINUE
        X1=L1*XP1+L2*YP1+XM0
        Y1=M1*XP1+M2*YP1+YM0
        Z1=N1*XP1+N2*YP1+ZM0
        X2=L1*XP2+L2*YP2+XM0
        Y2=M1*XP2+M2*YP2+YM0
        Z2=N1*XP2+N2*YP2+ZM0
 9      CONTINUE
  20    CONTINUE
  24    CONTINUE
        ISDTCH=0
        D12=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
        KINT=0
        IPPRL=0
        IF(JOP.EQ.1)CALL ZWTPE2(X1,Y1,Z1,X2,Y2,Z2,D12,2,
     2-1,XN1,YN1,ZN1,
     3 XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,YN4,ZN4,IN12,
     4 NINT,IACN,Z,KINT)
        IF(JOP.EQ.2)CALL ZWTDE(X1,Y1,Z1,X2,Y2,Z2,-1,XN1,YN1,ZN1,
     & XN2,YN2,ZN2,XN3,YN3,ZN3,NINT,BN,Z)
        IF(JOP.EQ.3)CALL ZWTWE(X1,Y1,Z1,X2,Y2,Z2,-1,XN1,YN1,ZN1,
     & XN2,YN2,ZN2,IN12,Z,0)
        IF(IFGD.EQ.4)WRITE(6,*)I,Z
        ZMN=ZMN+Z*W
10      CONTINUE
        ZMN=ZMN/(3.*NDT)
        RETURN
        END
```

APPENDIX 25

SUBROUTINE ZATAT2

```
      SUBROUTINE ZATAT2(B,H,Z,NL,ZS,ALFD)
      COMPLEX Z,CGDS,SGDS,SGDW,CGDW,ETA,GAM,P11,DUM,EGD
      COMPLEX ZWW,ZWD,ZDW,ZDD
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
C COMPUTE D/D,D/W,W/D
      DS=B-A
      EGD=CEXP(GAM*DS)
      CGDS=(EGD+1./EGD)/2.
      SGDS=(EGD-1./EGD)/2.
      EGD=CEXP(GAM*H)
      SGDW=(EGD-1./EGD)/2.
      CGDW=(EGD+1./EGD)/2.
      N=2*((NL+1)/2)
      DPH=2.0*PI/N
      ALF=ALFD*PI/180.0
      ZDD=(0.,0.)
      ZDW=(0.,0.)
      ZWD=(0.,0.)
      ZWW=(0.,0.)
      DO 10 I=1,N
      PH=(I-.5)*DPH
      CPSI=COS(PH)
      CALL GGMM1(A,B,A,B,Q,CGDS,SGDS,SGDS,CPSI,ETA,GAM,
     & P11,DUM,DUM,DUM)
      ZDD=ZDD+P11
      XA=A*COS(PH)
      YA=A*SIN(PH)
      ZA=0.0
      XB=B*COS(PH)
      YB=B*SIN(PH)
      ZB=0.0
      X1=A
      Y1=0.0
      Z1=0.0
      X2=A
      Y2=H*SIN(ALF)
      Z2=H*COS(ALF)
      CALLGGS1(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,Q,DS,CGDS,
     2SGDS,H,SGDW,0,ETA,GAM,P11,DUM,DUM,DUM)
      ZDW=ZDW-P11
      CALLGGS1(X1,Y1,Z1,X2.Y2,Z2,XA,YA,ZA,XB,YB,ZB,Q,H,CGDW,
     2SGDW,DS.SGDS,0,ETA,GAM,P11,DUM,DUM,DUM)
      ZWD=ZWD-P11
      D=A*SQRT(2.-2.*COS(PH))
      CALL GGMM1(0.,H,0.,H,D,CGDW,SGDW,SGDW,1.,ETA,GAM,
     &P11,DUM,DUM,DUM)
      ZWW=ZWW+P11
   10 CONTINUE
      ZDD=ZDD/N
      ZDW=ZDW/N
      ZWD=ZWD/N
      ZWW=ZWW/N
      Z=ZDD+ZWD+ZDW+ZWW
      Z=Z+(SGDW*CGDW-GAM*H)*ZS/(4.*PI*GAM*A*SGDW**2)
      RETURN
      END
```

APPENDIX 26

SUBROUTINE PDPZ

```
       SUBROUTINE PDPZ(XM0,YM0,ZM0,NAT,XN1,YN1,ZN1,XN2,YN2,ZN2,
      & XN3,YN3,ZN3,IN12,INTP,ERVSR,IAT,RMIN,DR,ZMN,DIST)
       COMPLEX ERVSR(IAT,400),ZMN.GAM,ETA,ER
       COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
       D=SQRT(((XN1+XN3)/2.-XM0)**2+((YN1+YN3)/2.-YM0)**2
      & +((ZN1+ZN3)/2.-ZM0)**2)
       NPE=INTP
       IF(D.GT. .5*WV) NPE=2*(NPE/6)
       IF(NPE.LT.2)NPE=2
       DX12=(XN2-XN1)/NPE
       DY12=(YN2-YN1)/NPE
       DZ12=(ZN2-ZN1)/NPE
       DX23=(XN3-XN2)/NPE
       DY23=(YN3-YN2)/NPE
       DZ23=(ZN3-ZN2)/NPE
       WN=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
       HN=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
       DW=WN/NPE
       DH=HN/NPE
       NPE1=NPE+1
       ZMN=(0.,0.)
       DO 10 I=1,NPE1
       W=3+(-1)**I
       IF(I.EQ.1 .OR. I.EQ.NPE1) W=W/2.
       YP=-WN/2.+(I-1)*DW
       DO 10 J=1.NPE1
       V=3+(-1)**J
       IF(J.EQ.1 .OR. J.EQ.NPE1) V=V/2.
       ZP=-HN/2.+(J-1)*DH
       X=XN1+(J-1)*DX12+(I-1)*DX23
       Y=YN1+(J-1)*DY12+(I-1)*DY23
       Z=ZN1+(J-1)*DZ12+(I-1)*DZ23
       RCAP=SQRT((XM0-X)**2+(YM0-Y)**2+(ZM0-Z)**2+Q**2)
       R=SQRT(ABS(RCAP**2-DIST*DIST))
       IF(R.LT.1.E-10) GOTO 10
       N=ABS(R-RMIN)/DR+1
       ER=ERVSR(NAT,N)+(ERVSR(NAT,N+1)-ERVSR(NAT,N))/DR*(R-RMIN-(N-1)*DR)
       ER=ER*((X-XM0)*DX12+(Y-YM0)*DY12+(Z-ZM0)*DZ12)*NPE*IN12
      & /(R*HN)
       ZMN=ZMN+ER*W*V*SIN(XK*(HN/2.-ZP))*COS(XK*YP)
10     CONTINUE
       ZMN=-ZMN*DH*DW*XK/(18.*SIN(XK*HN)*SIN(XK*WN/2.))
       RETURN
       END
```

APPENDIX 27

SUBROUTINE PDPZ1

```
      SUBROUTINE PDPZ1(XM0,YM0,ZM0,NAT,XN1,YN1,ZN1,XN2,YN2,ZN2,
   &  XN3,YN3,ZN3,XN4,YN4,ZN4,IACN,IN12,NPE,ERVSR,IAT,RMIN,
   &  DR,ZMN,DIST)
      COMPLEX ERVSR(IAT,500),ZMN.ZMNL,GAM,ETA,ER
      COMMON /A/ WV,PI,A,Q,GAM,ETA,XK
      IPLOT=0
      IF(IPLOT.LE.0)GO TO 15
      CALL PLOTS(0,0,0)
      CALL PLOT(4.25,5.0,-3)
      CALL PLOT(XN1,YN1,3)
      CALL PLOT(XN2,YN2,2)
      CALL PLOT(XN3,YN3,2)
      CALL PLOT(XN4,YN4,2)
      CALL PLOT(XN1,YN1,2)
      CALL SYMBOL(XM0,YM0,0.25,10.0.0,-1)
15    CONTINUE
      D=SQRT(((XN1+XN3)/2.-XM0)**2+((YN1+YN3)/2.-YM0)**2
   &  +((ZN1+ZN3)/2.-ZM0)**2)
      IF(D.GT.0.5*WV)NPE=2*(NPE/6)
      IF(NPE.LT.2)NPE=2
      IF(IACN.LE.0)GO TO 30
      DN14=SQRT((XN4-XN1)**2+(YN4-YN1)**2+(ZN4-ZN1)**2)
      DN23=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
      DMMX=AMAX1(DN14,DN23)
30    CONTINUE
      DX12=(XN2-XN1)/NPE
      DY12=(YN2-YN1)/NPE
      DZ12=(ZN2-ZN1)/NPE
      DX43=(XN3-XN4)/NPE
      DY43=(YN3-YN4)/NPE
      DZ43=(ZN3-ZN4)/NPE
      WN=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
      DW=WN/NPE
      NPE1=NPE+1
      ZMN=(0.,0.)
      FWT=0.0
      DO 10 I=1,NPE1
      W=3+(-1)**I
      IF(I.EQ.1 .OR. I.EQ.NPE1) W=W/2.
      XA=XN1+(I-1)*DX12
      YA=YN1+(I-1)*DY12
      ZA=ZN1+(I-1)*DZ12
      XB=XN4+(I-1)*DX43
      YB=YN4+(I-1)*DY43
      ZB=ZN4+(I-1)*DZ43
      HN=SQRT((XB-XA)**2+(YB-YA)**2+(ZB-ZA)**2)
      SINHN=SIN(XK*HN)
      DH=HN/NPE
      DXAB=(XB-XA)/NPE
      DYAB=(YB-YA)/NPE
      DZAB=(ZB-ZA)/NPE
      ZMNL=(.0,.0)
      DO 9 J=1,NPE1
      V=3+(-1)**J
      IF(J.EQ.1 .OR. J.EQ.NPE1) V=V/2.
      ZP=-HN/2.+(J-1)*DH
      X=XA+(J-1)*DXAB
      Y=YA+(J-1)*DYAB
      Z=ZA+(J-1)*DZAB
      IF(IPLOT.GT.0)CALL SYMBOL(X,Y,0.1,11,0.0,-1)
      RCAP=SQRT((XM0-X)**2+(YM0-Y)**2+(ZM0-Z)**2+Q**2)
      R=SQRT(ABS(RCAP**2-DIST*DIST))
      IF(R.LT.1.E-10) GO TO 9
      N=ABS(R-RMIN)/DR+1
      ER=ERVSR(NAT,N)+(ERVSR(NAT,N+1)-ERVSR(NAT,N))/DR*(R-RMIN-(N-1)*DR)
      ER=ER*((X-XM0)*DXAB+(Y-YM0)*DYAB+(Z-ZM0)*DZAB)*NPE
     & /(R*HN)
      ZMNL=ZMNL+ER*V*SIN(XK*(HN/2.-ZP))/WN
   9  CONTINUE
      ZMN=ZMN+ZMNL*W/SINHN
      FWT=FWT+W
10    CONTINUE
      ZMN=-(ZMN/2.)*IN12*DW/(3.*FWT)
      IF(IPLOT.GT.0) CALL PLOT(0.,0.,999)
      RETURN
      END
```

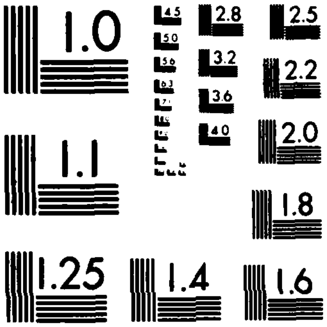
APPENDIX 28

SUBROUTINE ERDSK

```
      SUBROUTINEERDSK(A,B,X,Z,ETA,WV,NNPTS,EX)
      COMPLEXEX,J,E1,E2,E1R,E2R,C,ERL,ELL,ETA
      J=(0.0,1.0)
      PI=3.14159
      D=B-A
      XK=2.0*PI/WV
      DK=D*XK
      SDK=SIN(DK)
      CDK=COS(DK)
      C1=1.0/(2.0*PI)
      C=-J*ETA/(4.0*PI*SDK)
      NP1=NNPTS+1
      R=SQRT(X*X+Z*Z)
      EX=(0.0,0.0)
      H=2.0*PI/NNPTS
      DO100N=1,NP1
      PH=H/2.0+(N-1)*H
      CPH=COS(PH)
      SPH=SIN(PH)
      RH=X*CPH
      XM=RH*CPH
      YM=RH*SPH
      DM=SQRT((X-XM)**2+(YM)**2+Z**2)
      XA=A*CPH
      YA=A*SPH
      R1=SQRT((X-XA)**2+(YA)**2+Z**2)
      XB=B*CPH
      YB=B*SPH
      R2=SQRT((X-XB)**2+(YB)**2+Z**2)
      CT1=(RH-A)/R1
      CT2=(RH-B)/R2
      DX=(X-XM)/DM
      E1=CEXP(-J*XK*R1)
      E2=CEXP(-J*XK*R2)
      E1R=E1/R1
      E2R=E2/R2
      ERL=(J*E1*SDK+E1*CDK*CT1-E2*CT2)/DM
      ELL=E2R-E1R*CDK
      F=3+(-1)**N
      IF(N.EQ.1.OR.N.EQ.NP1)F=F/2.
      EX=EX+F*(ERL*DX+ELL*CPH)
  100 CONTINUE
      EX=-C1*C*H*EX/3.
      RETURN
      END
```

AD-A135 837 A USER'S MANUAL FOR ELECTROMAGNETIC SURFACE PATCH (ESP) 4/4
CODE VERSION II P..(U) OHIO STATE UNIV COLUMBUS
ELECTROSCIENCE LAB E H NEWMAN ET AL. SEP 83
UNCLASSIFIED ESL-712692-3 N00014-78-C-0049 F/G 20/14 NL

END
FILMED
1-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

APPENDIX 29

SUBROUTINE ZGSMM

```
      SUBROUTINE ZGSMM(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,
     & Y2,Z2,A,D1,CGD1,SGD1,D2,SGD2,Z12)
      COMMON /A/ WV,PI,AAA,Q,GAM,ETA,XK
      COMPLEX EJKR,Z12,P11,P12,P21,P22,GAM,ETA
      COMPLEX CGD1,SGD1,SGD2,FFF,FT,CST,T1,T2,T3,EJKT,FR
      R=SQRT((XA+XB-X1-X2)**2+(YA+YB-Y1-Y2)**2+
     &(ZA+ZB-Z1-Z2)**2)/2.
      INT=0
      IF(R.LT. .15*WV) GOTO 50
      RK=R*XK
      CAT=(X2-X1)/D2
      CBT=(Y2-Y1)/D2
      CGT=(Z2-Z1)/D2
      CAS=(XB-XA)/D1
      CBS=(YB-YA)/D1
      CGS=(ZB-ZA)/D1
      CAR=(X1+X2-XA-XB)/(2.*R)
      CBR=(Y1+Y2-YA-YB)/(2.*R)
      CGR=(Z1+Z2-ZA-ZB)/(2.*R)
      SDT=CAS*CAT+CBS*CBT+CGS*CGT
      CTH1=CAS*CAR+CBS*CBR+CGS*CGR
      CTH2=-(CAT*CAR+CBT*CBR+CGT*CGR)
      SS1=1.-CTH1*CTH1
      SS2=1.-CTH2*CTH2
      AB=ABS(SS1*SS2)
      IF(AB.LT. .8 .AND. R.LT. 1.2*(D1+D2)) GOTO 50
      INT=2
      GOTO 50
11    CONTINUE
      SDK1=+AIMAG(SGD1)
      SDK2=+AIMAG(SGD2)
      CDK1=REAL(CGD1)
      CDK2=SQRT(1.-SDK2*SDK2)
      R2=RK*RK
      EJKR=CMPLX(COS(RK),-SIN(RK))
      IF(AB.GT..001) GOTO 30
      Z12=FR*EJKR
      GOTO 60
30    T1=(0.,1.)*CTH1*SDK1-CDK1
      EJKT=CEXP((0.,1.)*XK*D1*CTH1/2.)
      T2=(0.,-1.)*((XK*D1/2.*T1+SDK1))*EJKT+(0.,1.)*XK*D1/(2.*EJKT)
      T2=T2/(2.*SDK1)
      T3=(0.,1.)*(XK*XK*D1*D1*SDK1/2.+XK*XK*D1*D1/4.+XK**3*D1**3/8.*T1)
      T3=T3*EJKT-(0.,1.)*XK**3*D1**3/(8.*EJKT)
      T3=T3/(2.*SDK1)
      T3=(0.,0.)
      FR=(0.,-1.)*XK*T2+(0.,1.)*T3/(XK*R*R)
      FR=FR*(0.,-60.)*EJKR/(R2)*FFF(SS2,CTH2,XK,D2,SDK2,CDK2)
      FT=FFF(SS1,CTH1,XK,D1,SDK1,CDK1)*FFF(SS2,CTH2,XK,D2,SDK2,CDK2)
      FT=FT*(SDT+CTH1*CTH2)
      CST=30.*XK*XK*EJKR/RK
      Z12=CST*FT*CMPLX(1./RK,1.-1./R2)
      Z12=Z12-CTH2*FR
      GOTO 60
50    CONTINUE
102   FORMAT(' INT/MM=',I2)
      CALL GGS1(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,
     & A,D1,CGD1,SGD1,D2,SGD2.INT,ETA,GAM,P11,P12,P21,P22)
      Z12=P11
60    RETURN
      END
      FUNCTION FFF(SST,CTH,XK,D,SDK,CDK)
      COMPLEX FFF,EJK
      EJK=CEXP((0.,1.)*XK*CTH*D/2.)
      IF(SST.LT. .001) GOTO 10
      FFF=EJK*((0.,1.)*CTH*SDK-CDK)+1./EJK
      FFF=FFF/(XK*SST*SDK)
      RETURN
10    FFF=(0.,1.)*(XK*D/EJK-SDK*EJK)/(2.*XK*SDK)
      RETURN
      END
```

APPENDIX 30

SUBROUTINE GGS1

```
      SUBROUTINE GGS1(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AM
     2,DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,P11,P12,P21,P22)
      COMPLEX P11,P12.P21,P22,EJA,EJB,EJ1,EJ2,ETA,GAM,C1,C2,CST
      COMPLEX EGD,CGDS.SGDS,SGDT,ER1,ER2,ET1,ET2
      DATA FP/12.56637/
      CA=(X2-X1)/DT
      CB=(Y2-Y1)/DT
      CG=(Z2-Z1)/DT
      CAS=(XB-XA)/DS
      CBS=(YB-YA)/DS
      CGS=(ZB-ZA)/DS
      CC=CA*CAS+CB*CBS+CG*CGS
      IF(ABS(CC).GT..997)GO TO 200
   20 SZ=(X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      IF(INT.LE.0)GO TO 300
      INS=2*(INT/2)
      IF(INS.LT.2)INS=2
      IP=INS+1
      DELT=DT/INS
      T=.0
      DSZ=CC*DELT
      P11=(.0,.0)
      P12=(.0,.0)
      P21=(.0,.0)
      P22=(.0,.0)
      AMS=AM*AM
      SGN=-1.
      DO 100 IN=1.IP
      ZZ1=SZ
      ZZ2=SZ-DS
      XXZ=X1+T*CA-XA-SZ*CAS
      YYZ=Y1+T*CB-YA-SZ*CBS
      ZZZ=Z1+T*CG-ZA-SZ*CGS
      RS=XXZ**2+YYZ**2+ZZZ**2
      R1=SQRT(RS+ZZ1**2)
      EJA=CEXP(-GAM*R1)
      EJ1=EJA/R1
      R2=SQRT(RS+ZZ2**2)
      EJB=CEXP(-GAM*R2)
      EJ2=EJB/R2
      ER1=EJA*SGDS+ZZ1*EJ1*CGDS-ZZ2*EJ2
      ER2=-EJB*SGDS+ZZ2*EJ2*CGDS-ZZ1*EJ1
      FAC=.0
      IF(RS.GT.AMS)FAC=(CA*XXZ+CB*YYZ+CG*ZZZ)/RS
      ET1=CC*(EJ2-EJ1*CGDS)+FAC*ER1
      ET2=CC*(EJ1-EJ2*CGDS)+FAC*ER2
      C=3.+SGN
      IF(IN.EQ.1 .OR. IN.EQ.IP)C=1.
      EGD=CEXP(GAM*(DT-T))
      C1=C*(EGD-1./EGD)/2.
      EGD=CEXP(GAM*T)
      C2=C*(EGD-1./EGD)/2.
      P11=P11+ET1*C1
      P12=P12+ET1*C2
      P21=P21+ET2*C1
      P22=P22+ET2*C2
      T=T+DELT
      SZ=SZ+DSZ
  100 SGN=-SGN
      CST=-ETA*DELT/(3.*FP*SGDS*SGDT)
      P11=CST*P11
      P12=CST*P12
      P21=CST*P21
      P22=CST*P22
      RETURN
  200 SZ1=(X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      RH1=SQRT((X1-XA-SZ1*CAS)**2+(Y1-YA-SZ1*CBS)**2+(Z1-ZA-SZ1*CGS)**2)
      SZ2=SZ1+DT*CC
      RH2=SQRT((X2-XA-SZ2*CAS)**2+(Y2-YA-SZ2*CBS)**2+(Z2-ZA-SZ2*CGS)**2)
      DDD=(RH1+RH2)/2.
      IF(DDD.GT.20.*AM .AND. INT.GT.0)GO TO 20
      IF(DDD.LT.AM)DDD=AM
      CALL GGMM(.0,DS,SZ1,SZ2.DDD,CGDS,SGDS,SGDT,1.
     2,ETA,GAM,P11.P12,P21,P22)
      RETURN
  300 SS=SQRT(1.-CC*CC)
      CAD=(CGS*CB-CBS*CG)/SS
      CBD=(CAS*CG-CGS*CA)/SS
      CGD=(CBS*CA-CAS*CB)/SS
      DK=(X1-XA)*CAD+(Y1-YA)*CBD+(Z1-ZA)*CGD
      DK=ABS(DK)
      IF(DK.LT.AM)DK=AM
      XZ=XA+SZ*CAS
      YZ=YA+SZ*CBS
      ZZ=ZA+SZ*CGS
      XP1=X1-DK*CAD
      YP1=Y1-DK*CBD
      ZP1=Z1-DK*CGD
      CAP=CBS*CGD-CGS*CBD
      CBP=CGS*CAD-CAS*CGD
      CGP=CAS*CBD-CBS*CAD
      P1=CAP*(XP1-XZ)+CBP*(YP1-YZ)+CGP*(ZP1-ZZ)
      T1=P1/SS
      S1=T1*CC-SZ
      CALL GGMM(S1,S1+DS,T1,T1+DT,DK,CGDS,SGDS,SGDT,CC,ETA,GAM
     2,P11,P12,P21,P22)
      RETURN
      END
```

APPENDIX 31

SUBROUTINE COUPLE

```
      SUBROUTINECOUPLE(ZT,ZTF,M1,M2,SN1,SN2,I12,V,NT,IFIL,ICC)
      COMPLEXZT(1),V(1),D,Y11,Y12,Y21,Y22,Z11,Z12,Z21,Z22
      COMPLEX ZTF(ICC,ICC)
      DO100I=1,2
      I12=I
      M=M1
      IF(I.EQ.2)M=M2
      DO110K=1,NT
  110 V(K)=(0.0,0.0)
      V(M)=(1.0,0.0)
      IF(IFIL.EQ.0)CALL SQROT(ZT,V,0,I12,NT)
      IF(IFIL.EQ.1)CALL CROUT(ZTF,V,ICC,0,0,I12,NT)
      IF(I.EQ.2)GOTO120
      Y11=V(M1)
      Y12=V(M2)
      GOTO100
  120 Y22=V(M2)
      Y21=V(M1)
  100 CONTINUE
      Y11=Y11*SN1*SN1
      Y21=Y21*SN1*SN2
      Y22=Y22*SN2*SN2
      Y12=Y12*SN2*SN1
C     WRITE(6,*)Y11,Y12
C     WRITE(6,*)Y21,Y22
      D=Y11*Y22-Y12*Y21
      Z11=Y22/D
      Z12=-Y12/D
      Z21=-Y21/D
      Z22=Y11/D
      WRITE(6,130)
  130 FORMAT(//3X,'TWO PORT IMPEDANCE MATRIX (OHMS)'/)
      WRITE(6,140)Z11,Z12,Z21,Z22
  140 FORMAT(3X,'Z11 = ',2E13.3/3X,'Z12 = ',2E13.3/3X,'Z21 = ',
     2 2E13.3/3X,'Z22 = ',2E13.3/)
      EL=CABS(Y12*Y21)/(2.0*REAL(Y11)*REAL(Y22)-REAL(Y12*Y21))
      EL=ABS(EL)
      GMAX=EL/2.0
      IF(ABS(EL).GT.0.003)GMAX=(1.0-SQRT(1.0-EL*EL))/EL
      GMAX=10.0*ALOG10(GMAX)
      WRITE(6,200)GMAX
  200 FORMAT(3X,'MAX. COUPLING IN DB = ',F10.3/)
      TYPE*,GMAX
      RETURN
      END
```

APPENDIX 32

SUBROUTINE ANTV

```
      SUBROUTINE ANTV(I1,I2,I3,IA,IB,IWR,JA,JB,NM,
     &ZT,CJ,CG,VG,Y11,Z11,NWR,NPL,NAT,VGA,PIN,
     2AM,CMM,D,DISS,GAM,SGD,ZLD,ZS,ZLDA,INM,MD,ND,NSA)
      COMPLEX CJ(1),VG(1),Y11,Z11,CG(1)
      COMPLEX VGA(1),ZT(1),GAM,SGD(1),ZS,ZLDA(1),ZLD(1)
      DIMENSION I1(1),I2(1),I3(1),IA(1),IB(1),JA(1),JB(1)
      DIMENSIOND(1),MD(INM,4),ND(1),NSA(1)
      IJ(I,J,N)=(J-1)*N-(J*J-J)/2+I
      NTOT=NWR+NPL+NAT
      DO 50 I=1,NTOT
      CJ(I)=(.0,.0)
      IF(I.GT.NWR)GOTO50
      K=JA(I)
      DO 40 KK=1,2
      KA=IA(K)
      KB=IB(K)
      JJ=K
      FI=1.
      IF(KB.EQ.I2(I))GO TO 36
      IF(KB.EQ.I1(I))FI=-1.
      CJ(I)=CJ(I)+FI*VG(JJ)
      GO TO 40
   36 IF(KA.EQ.I3(I))FI=-1.
      JJ=K+NM
      CJ(I)=CJ(I)+FI*VG(JJ)
   40 K=JB(I)
   50 CONTINUE
      IF(NAT.EQ.0) GOTO 89
      DO90I=1,NAT
      K=NWR+NPL+I
   90 CJ(K)=VGA(I)
 89   CONTINUE
 60   FORMAT(//3X,'LISTING OF GENERATORS'/)
      DO 55 I=1,NTOT
      IF(CABS(CJ(I)).EQ.0.) GOTO 55
      IF(I.GT.NWR+NPL) GOTO 65
 C    WRITE(6,70) CJ(I),I
 70   FORMAT(3X,2F8.3,2X,'VOLT GENERATOR IN WIRE MODE',I3)
      GOTO 55
 65   II=I-NWR-NPL
```

```
C      WRITE(6,75) CJ(I),II
75     FORMAT(3X,2F8.3,2X,'VOLT GENERATOR IN ATTACHMENT MODE',I3)
  55   CG(I)=CJ(I)
102    CALL SQROT(ZT,CJ,IWR,1,NTOT)
       Y11=(.0,.0)
       DO 80 I=1,NTOT
  80   Y11=Y11+CJ(I)*CONJG(CG(I))
       Z11=1./Y11
       PIN=REAL(Y11)
       CALLARITE(IA,IB,INM,0.I1,I2,I3,MD,ND,NM,CJ,CG,NSA,NWR,NPL,NAT)
       CALLAGDISS(AM,CG,CMM,D,DISS,GAM,NM,SGD,ZLD,ZS,ZLDA,NAT,NSA)
       PRAD=PIN-DISS
       EFF=100.0*PRAD/PIN
       WRITE(6,513) Y11,Z11,EFF
  513  FORMAT(/3X,'INPUT ADMITTANCE(MHOS) = ',F10.6,' J ',F10.6/
      23X,'INPUT IMPEDANCE(OHMS) = ',F10.3,' J ',F10.3/,
      33X,'EFFICIENCY(PERCENT) = ',F7.3/)
       RETURN
       END
```

APPENDIX 33

SUBROUTINE CROUT

```
      SUBROUTINE CROUT(C,S.ICC,ISYM,IWR.I12,N)
      COMPLEX C(ICC,ICC),S(1)
      COMPLEX F,P,SS,T
    2 FORMAT(1X,1I5,1F10.3,1F15.7,1F10.0)
    5 FORMAT(1H0)
      IF(I12.NE.1)GO TO 22
      IF(N.EQ.1)S(1)=S(1)/C(1,1)
      IF(N.EQ.1)GO TO 100
      IF(ISYM.NE.0)GO TO 8
      DO 6 I=1,N
      DO 6 J=I,N
    6 C(J,I)=C(I,J)
    8 F=C(1,1)
      DO 10 L=2,N
   10 C(1,L)=C(1,L)/F
      DO 20 L=2,N
      LLL=L-1
      DO 20 I=L,N
      F=C(I,L)
      DO 11 K=1,LLL
   11 F=F-C(I,K)*C(K,L)
      C(I,L)=F
      IF(L.EQ.I)GO TO 20
      P=C(L,L)
      IF(ISYM.EQ.0)GO TO 15
      F=C(L,I)
      DO 12 K=1,LLL
   12 F=F-C(L,K)*C(K,I)
      C(L,I)=F/P
      GO TO 20
   15 F=C(I,L)
      C(L,I)=F/P
   20 CONTINUE
   22 DO 30 L=1,N
      P=C(L,L)
      T=S(L)
      IF(L.EQ.1)GO TO 30
      LLL=L-1
      DO 25 K=1,LLL
   25 T=T-C(L,K)*S(K)
```

```
 30  S(L)=T/P
     DO 38 L=2,N
     I=N-L+1
     II=I+1
     T=S(I)
     DO 35 K=II,N
 35  T=T-C(I,K)*S(K)
 38  S(I)=T
     IF(IWR.LE.0) GO TO 100
     WRITE(6,5)
     CNOR=.0
     DO 40 I=1,N
     SA=CABS(S(I))
 40  IF(SA.GT.CNOR)CNOR=SA
     IF(CNOR.LE.0.)CNOR=1.
     DO 44 I=1,N
     SS=S(I)
     SA=CABS(SS)
     SNOR=SA/CNOR
     PH=.0
     IF(SA.GT.0.)PH=57.29578*ATAN2(AIMAG(SS),REAL(SS))
 44  WRITE(6,2)I,SNOR,SA,PH
     WRITE(6,5)
100  RETURN
     END
```

APPENDIX 34

SUBROUTINE AGDISS

```
      SUBROUTINEAGDISS(AM,CG,CMM,D,DISS,GAM,NM,SGD,ZLD,ZS,ZLDA,NAT,NSA)
      COMPLEX CG(1),SGD(1),ZLD(1),CJA,CJB,GAM,ZS
      COMPLEXZLDA(1),ZJ,ZK
      DIMENSION D(1),NSA(1)
      DATA PI/3.14159/
      DISS=.0
      IF(CMM.LE.0.)GO TO 120
      ALPH=REAL(GAM)
      BETA=AIMAG(GAM)
      RH=REAL(ZS)/(4.*PI*AM)
      DO 100 K=1.NM
      DK=D(K)
      DEN=CABS(SGD(K))**2
      EAD=EXP(ALPH*DK)
      CAD=(EAD+1./EAD)/2.
      CBD=COS(BETA*DK)
      SAD=DK
      IF(ALPH.NE.0.)SAD=(EAD-1./EAD)/(2.*ALPH)
      SBD=DK
      IF(BETA.NE.0.)SBD=SIN(BETA*DK)/BETA
      FA=RH*(SAD*CAD-SBD*CBD)/DEN
      FB=2.*RH*(CAD*SBD-SAD*CBD)/DEN
      CJA=CG(K)
      L=K+NM
      CJB=CG(L)
100   DISS=DISS+FA*(CABS(CJA)**2+CABS(CJB)**2)
    2 +FB*(REAL(CJA)*REAL(CJB)+AIMAG(CJA)*AIMAG(CJB))
120   DO 140 J=1,NM
      K=J+NM
      ZJ=ZLD(J)
      ZK=ZLD(K)
      IF(NAT.EQ.0)GOTO150
      DO160NA=1,NAT
      IF(NSA(NA).EQ.J)ZJ=ZJ+ZLDA(NA)
      IF(NSA(NA).EQ.K)ZK=ZK+ZLDA(NA)
160   CONTINUE
150   CONTINUE
140   DISS=DISS+REAL(ZJ)*(CABS(CG(J))**2)
    2 +REAL(ZK)*(CABS(CG(K))**2)
      RETURN
      END
```

APPENDIX 35

SUBROUTINE ARITE

```
      SUBROUTINE ARITE(IA,IB,INM,IWR,I1,I2,I3,MD,ND,NM, [illegible],CG,
    2 NSA,NWR,NPLTM,NAT)
      COMPLEX CJ(1),CG(1),CJA,CJB
      DIMENSION IA(1),IB(1),I1(1),I2(1),I3(1),MD(INM,4 [illegible]D(1),NSA(1)
2     FORMAT(1X,1I5,2G10.3.2F10.1,4G15.6)
5     FORMAT(1H0)
      AMAX=.0
      DO 100 K=1,NM
      KA=IA(K)
      KB=IB(K)
      CJA=(.0,.0)
      CJB=(.0,.0)
      NDK=ND(K)
      IF(NAT.EQ.0)GOTO150
      DO160NA=1,NAT
      NN=NWR+NPLTM+NA
      IF(NSA(NA).EQ.K)CJA=CJ(NN)
      IF(NSA(NA).EQ.K+NM)CJB=-CJ(NN)
160   CONTINUE
150   CONTINUE
      DO 40 II=1,NDK
      I=MD(K,II)
      FI=1.
      IF(KB.EQ.I2(I))GO TO 36
      IF(KB.EQ.I1(I))FI=-1.
      CJA=CJA+FI*CJ(I)
      GO TO 40
36    IF(KA.EQ.I3(I))FI=-1.
      CJB=CJB+FI*CJ(I)
40    CONTINUE
      CG(K)=CJA
      KK=K+NM
      CG(KK)=CJB
      ACJ=CABS(CJA)
      BCJ=CABS(CJB)
      IF(ACJ.GT.AMAX)AMAX=ACJ
      IF(BCJ.GT.AMAX)AMAX=BCJ
100   CONTINUE
      IF(IWR.GT.0)GO TO 110
      RETURN
```

```
110   IF(AMAX.LE.0.)AMAX=1.
      DO 200 K=1,NM
      CJA=CG(K)
      KK=K+NM
      CJB=CG(KK)
      ACJ=CABS(CJA)/AMAX
      BCJ=CABS(CJB)/AMAX
      PA=.0
      PB=.0
      IF(ACJ.GT.0.)PA=57.29578*ATAN2(AIMAG(CJA),REAL(CJA))
      IF(BCJ.GT.0.)PB=57.29578*ATAN2(AIMAG(CJB),REAL(CJB))
200   WRITE(6,2)K,ACJ,BCJ,PA,PB,CJA,CJB
      WRITE(6,5)
      RETURN
      END
```

APPENDIX 36

SUBROUTINE SORTB

```
       SUBROUTINE SORTB(IA,IB,I1,I2,I3,NWR,NM,A,CGD,SGD,FHZ,D,
      &IWRSQ,I12,ISCAT,ZTF,ZT,IFIL,ICC,ETT,EPP,
      &X,Y,Z,NPL,NAT,PA,PB,NSA,NPLA,PCN.BDSK,IQUAD,
      &NPLTM,IPL,IPLM,CJP,CJT,ETTS,EPPS,ETPS,EPTS,THETA,PHI,JA,JB,
      &SCSP,SCST,SPPM,SPTM,STPM,STTM,IMAGE,ICN,NDNPLT)
       DIMENSIONIA(1),IB(1),I1(1),I2(1),I3(1),
      2D(1),X(1),Y(1),Z(1),IPN(1),IQUAD(1)
       DIMENSIONPA(IPLM,4,3),PB(IPLM,4,3),NSA(1),NPLA(1),PCN(3,ICN,IPL)
       DIMENSION NDNPLT(1),BDSK(1),NM12N(1),NM23N(1),JA(1),JB(1)
       COMPLEXSGD(1),CGD(1),ETTS,EPPS,ETPS,EPTS,ET,EP,DUM
       COMPLEXXJ,ETA,GAM,VP,VT,CJI
       COMPLEX ETT(1),EPP(1),ZT(1)
       COMPLEX CJP(1),CJT(1),ZTF(ICC,ICC),SGDN.CGDN
       COMPLEX CSUMP,CSUMT
       DATA PI,TP/3.1415926,6.283185/
       AB(I,A,B)=(I-1)*B+(2-I)*A
       NTOT=NWR+NPLTM+NAT
       XJ=CMPLX(0.0.1.0)
       WV=2.998E8/FHZ
       XK=2.0*PI/WV
       ETA=CMPLX(120.0*PI,0.0)
       GAM=XJ*XK
       CJI=-4.*PI/(ETA*GAM)
       GGG=REAL(1./ETA)
       ETTS=(0.,0.)
       EPPS=(0.,0.)
       IWR=0
       IF(ISCAT.NE.1)GO TO 15
       DO 10 I=1,NTOT
       CJP(I)=(.0,.0)
    10 CJT(I)=(.0,.0)
    15 IF(ISCAT.EQ.0)GO TO 20
       DO 12 I=1,NTOT
       ETT(I)=(.0,.0)
    12 EPP(I)=(.0,.0)
C
C
    20    CONTINUE
          IAT=1+IMAGE
          DO 200 IAS=1,IAT
```

```
        IF(IAS.EQ.2)GOTO210
        THT=THETA
        CTH=COS(THETA)
        STH=SIN(THETA)
        CPH=COS(PHI)
        SPH=SIN(PHI)
        PFAC=1.0
        GOTO220
  210   CONTINUE
        THT=PI-THETA
        CTH=COS(PI-THETA)
        STH=SIN(PI-THETA)
        PFAC=-1.0
  220   CONTINUE
      DO130N=1,NTOT
      DO160J=1,2
C     DETERMINE EXPANSION MODE TYPE.
C
      IF(N.GT.NWR+NPLTM)GOTO270
      IF (N.GT. NWR) GOTO 260
C EXPANSION MODE IS A WIRE
      L=N
      I1L=I2(L)
      I2L=I1(L)
      IF (J .EQ. 2) I2L=I3(L)
      XN1=X(I1L)
      YN1=Y(I1L)
      ZN1=Z(I1L)
      XN2=X(I2L)
      YN2=Y(I2L)
      ZN2=Z(I2L)
      IN12=(-1)**J
      LL=JA(L)
      IF(J .EQ. 2) LL=JB(L)
      DN=D(LL)
      CGDN=CGD(LL)
      SGDN=SGD(LL)
      CALL GFF(XN1,YN1,ZN1,XN2,YN2,ZN2,DN,CGDN,SGDN,CTH,STH,CPH,
     & SPH,GAM,ETA,ET,DUM,EP,DUM)
      EP=EP*PFAC
      IF(ISCAT.NE.0)GO TO 50
      IF(IWR.EQ.2)WRITE(6,*)N,J,XN1,YN1,ZN1,XN2,YN2,ZN2
      ET=ET*IN12
      EP=EP*IN12
      IF(IWR.GE.1) WRITE(6,*) N,J,ET,EP,CJP(N)
      ETTS=ETTS+ET*CJP(N)
      EPPS=EPPS+EP*CJP(N)
      GOTO 160
 50   IF(IAS.EQ.1)ETT(N)=ETT(N)+ET*IN12
      IF(IAS.EQ.1)EPP(N)=EPP(N)+EP*IN12
      IF(ISCAT.EQ.2)GO TO 160
      CJP(N)=CJP(N)+EP*CJI*IN12
      CJT(N)=CJT(N)+ET*CJI*IN12
      GO TO 160
 260  CONTINUE
C
C     EXPANSION MODE IS A PLATE.
C
```

```
      L=N-NWR
      IACN=IQUAD(L)
      IN12=(-1)**J
      XN1=AB(J,PA(L,1,1),PB(L,1,1))
      YN1=AB(J,PA(L,1,2),PB(L,1,2))
      ZN1=AB(J,PA(L,1,3),PB(L,1,3))
      XN2=AB(J,PA(L,2,1),PB(L,2,1))
      YN2=AB(J,PA(L,2,2),PB(L,2,2))
      ZN2=AB(J,PA(L,2,3),PB(L,2,3))
      XN3=AB(J,PA(L,3,1),PB(L,3,1))
      YN3=AB(J,PA(L,3,2),PB(L,3,2))
      ZN3=AB(J,PA(L,3,3),PB(L,3,3))
      XN4=AB(J,PA(L,4,1),PB(L,4,1))
      YN4=AB(J,PA(L,4,2),PB(L,4,2))
      ZN4=AB(J,PA(L,4,3),PB(L,4,3))
      IF(IACN.NE.-3)GO TO 11
      GO TO 14
11    NPLS=10
      IF(L.GT.NDNPLT(NPL))NPLS=6
      CALL SURFFP(XN4,YN4,ZN4,XN1,YN1,ZN1,XN2,YN2,ZN2.XN3,YN3,
     1ZN3,NPLS,GAM,ETA,XK,FHZ,IN12,THT,PHI,ET,EP)
      IF(IWR.EQ.2)WRITE(6,*)XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,
     1XN4,YN4.ZN4
      IF(IWR.GE.1)WRITE(6,*)N,J,ET,EP
      EP=EP*PFAC
      IF(ISCAT.NE.0)GO TO 101
      CJT(N)=CJP(N)
      ETTS=ETTS+ET*CJT(N)
      EPPS=EPPS+EP*CJP(N)
      GO TO 160
101   IF(IAS.EQ.1)ETT(N)=ETT(N)+ET
      IF(IAS.EQ.1)EPP(N)=EPP(N)+EP
      IF(ISCAT.EQ.2)GO TO 160
      CJP(N)=CJP(N)+EP*CJI
      CJT(N)=CJT(N)+ET*CJI
      GO TO 160
14    CALL SURMFF(XN1,YN1,ZN1.XN2,YN2,ZN2,XN3,YN3,ZN3.IN12.
     &THT,PHI,ET,EP,WV)
13    EP=EP*PFAC
      IF(ISCAT.NE.0)GO TO 100
      IF(IWR.EQ.2)WRITE(6,*)N.J,XN1.YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3
      IF(IWR.GE.1) WRITE(6,*)N.J,ET,EP
      ETTS=ETTS+ET*CJP(N)
      EPPS=EPPS+EP*CJP(N)
      GOTO160
  100 IF(IAS.EQ.1)ETT(N)=ETT(N)+ET
      IF(IAS.EQ.1)EPP(N)=EPP(N)+EP
      IF(ISCAT.EQ.2)GO TO 160
      CJP(N)=CJP(N)+EP*CJI
      CJT(N)=CJT(N)+ET*CJI
      GO TO 160
  270 CONTINUE
C
C     EXPANSION MODE IS AN ATTACHMENT MODE.
C
      L=N-NWR-NPLTM
      IF(J.EQ.2)GOTO290
C DISK COMPONENT OF ATTACHMENT MODE.
```

```
      NAS=NSA(L)
      IF(NAS.GT.NM)GOTO300
      NAP=IA(NAS)
      GOTO310
  300 CONTINUE
      NAS=NAS-NM
      NAP=IB(NAS)
  310 CONTINUE
      XN1=X(NAP)
      YN1=Y(NAP)
      ZN1=Z(NAP)
      NPLL=NPLA(L)
      XN2=PCN(1,1,NPLL)
      YN2=PCN(2,1,NPLL)
      ZN2=PCN(3,1,NPLL)
      XN3=PCN(1,2,NPLL)
      YN3=PCN(2,2,NPLL)
      ZN3=PCN(3,2.NPLL)
      B=BDSK(L)
      CALL DSKFF(XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,THT,PHI,
     & A,B,WV,ET,EP)
      EP=EP*PFAC
      IF(IWR.EQ.2)WRITE(6,*)N,J,XN1.YN1.ZN1,XN2,YN2.ZN2,XN3,YN3,ZN3
      IF(IWR.GE.1) WRITE(6,*)N,J,ET,EP
      IF(ISCAT.NE.0)GOTO100
      ETTS=ETTS+ET*CJP(N)
      EPPS=EPPS+EP*CJP(N)
      GOTO 160
  290 CONTINUE
C WIRE COMPONENT OF ATTACHMENT MODE.
      NAS=NSA(L)
      IF(NAS.GT.NM)GOTO320
      NPP=IB(NAS)
      GOTO330
  320 CONTINUE
      NAS=NAS-NM
      NPP=IA(NAS)
  330 CONTINUE
      XN2=X(NPP)
      YN2=Y(NPP)
      ZN2=Z(NPP)
      DN=D(NAS)
      CGDN=CGD(NAS)
      SGDN=SGD(NAS)
      CALL GFF(XN1,YN1,ZN1,XN2,YN2,ZN2.DN,CGDN.SGDN.CTH,STH,CPH.
     &SPH,GAM,ETA,ET,DUM,EP,DUM)
      EP=EP*PFAC
      IF(IWR.EQ.2)WRITE(6,*)N.J,XN1,YN1,ZN1,XN2,YN2,ZN2
      IF(IWR.GE.1) WRITE(6.*)N.J,ET,EP
      IF(ISCAT.NE.0)GOTO100
      ETTS=ETTS+ET*CJP(N)
      EPPS=EPPS+EP*CJP(N)
  160 CONTINUE
  130 CONTINUE
  200 CONTINUE
      IF(ISCAT.NE.1)GO TO 170
 135  FORMAT(//' CURRENTS FOR PHI POLARIZED PLANE WAVE')
 136  FORMAT(//' CURRENTS FOR THETA POLARIZED PLANE WAVE')
```

```
137  FORMAT(3X,I4,2(5X,2E20.5))
138  FORMAT(//'RIGHT-HANDSIDE VECTOR:  PHI POLARIZATION',30X,
    2'THETA POLARIZATION'/)
132  FORMAT(//'IMPEDANCE MATRIX:  UPPER TRIANGULAR MATRIX'/)
134  FORMAT(2X,2E14.4,4X,2E14.4,4X,2E14.4,4X,2E14.4,4X)
141  FORMAT(//'RIGHT-HANDSIDE VECTOR:  BY MULT. ZT*CJP,ZT*CJT')
     IF(IWRSQ.LE.0)GO TO 139
 133 CONTINUE
     DO 172 IJ=1,NTOT
172  CONTINUE
139  CONTINUE
     IF(IWRSQ.GE.1)WRITE(6,135)
     IF(IFIL.EQ.0) CALL SQROT(ZT,CJP,IWRSQ,I12,NTOT)
     IF(IFIL.EQ.1) CALL CROUT(ZTF,CJP,ICC,1,IWRSQ,I12,NTOT)
     I12=2
     IF(IWRSQ.GE.1)WRITE(6,136)
     IF(IFIL.EQ.0) CALL SQROT(ZT,CJT,IWRSQ,I12,NTOT)
     IF(IFIL.EQ.1) CALL CROUT(ZTF,CJT,ICC,1,IWRSQ,I12,NTOT)
     PIN=.0
     TIN=.0
     DO 164 I=1,NTOT
     VP=CJI*EPP(I)
     VT=CJI*ETT(I)
     PIN=PIN+REAL(VP*CONJG(CJP(I)))
 164 TIN=TIN+REAL(VT*CONJG(CJT(I)))
     ECSP=PIN/GGG
     ECST=TIN/GGG
     SCSP=ECSP
     SCST=ECST
 170 IF(ISCAT.EQ.0)RETURN
     EPTS=(.0,.0)
     ETPS=(.0,.0)
     DO 180 I=1,NTOT
     EPPS=EPPS+CJP(I)*EPP(I)
     EPTS=EPTS+CJP(I)*ETT(I)
     ETTS=ETTS+CJT(I)*ETT(I)
 180 ETPS=ETPS+CJT(I)*EPP(I)
     SPPM=2.*TP*(CABS(EPPS)**2)
     SPTM=2.*TP*(CABS(EPTS)**2)
     STPM=2.*TP*(CABS(ETPS)**2)
     STTM=2.*TP*(CABS(ETTS)**2)
     RETURN
     END
```

APPENDIX 37

SUBROUTINE SURFFP

```
      SUBROUTINE SURFFP(XN1,YN1,ZN1,XN2,YN2,ZN2,XN3,YN3,ZN3,XN4,
     2YN4,ZN4,NPLS,GAM,ETA,XK,FHZ,IN12,THR,PHR,ETH,EPH)
      COMPLEX ETH,EPH,ET1,EP1,DUM
      COMPLEX EGD,CGD,SGD,GAM,ETA
      WV=2.998E8/FHZ
      DX12=(XN2-XN1)/NPLS
      DY12=(YN2-YN1)/NPLS
      DZ12=(ZN2-ZN1)/NPLS
      DX43=(XN3-XN4)/NPLS
      DY43=(YN3-YN4)/NPLS
      DZ43=(ZN3-ZN4)/NPLS
      HH1=SQRT(DX12*DX12+DY12*DY12+DZ12*DZ12)
      NP1=NPLS+1
      CTH=COS(THR)
      STH=SIN(THR)
      CPH=COS(PHR)
      SPH=SIN(PHR)
      WNT=SQRT((XN2-XN1)**2+(YN2-YN1)**2+(ZN2-ZN1)**2)
      ETH=(.0,.0)
      EPH=(.0,.0)
      FF=0.0
      DD14=SQRT((XN4-XN1)**2+(YN4-YN1)**2+(ZN4-ZN1)**2)
      DD23=SQRT((XN3-XN2)**2+(YN3-YN2)**2+(ZN3-ZN2)**2)
      DDMX=MAX(DD14,DD23)
      DO 10 IS=1,NP1
      W=3.+(-1)**IS
      IF(IS.EQ.1.OR.IS.EQ.NP1)W=W/2.0
      X1=XN1+(IS-1)*DX12
      Y1=YN1+(IS-1)*DY12
      Z1=ZN1+(IS-1)*DZ12
      X2=XN4+(IS-1)*DX43
      Y2=YN4+(IS-1)*DY43
      Z2=ZN4+(IS-1)*DZ43
16    DD=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
      IF(DD.GT.0.0001*WV)GO TO 17
      SFB=1.0
      IF(IS.EQ.1)SFB=-1.0
      X1=X1-0.01*DX12*SFB
      Y1=Y1-0.01*DY12*SFB
      Z1=Z1-0.01*DZ12*SFB
      X2=X2-0.01*DX43*SFB
      Y2=Y2-0.01*DY43*SFB
      Z2=Z2-0.01*DZ43*SFB
      GO TO 16
 17   CONTINUE
      EGD=CEXP(GAM*DD)
      SGD=(EGD-1./EGD)/2.0
      CGD=(EGD+1./EGD)/2.0
      CALL GFF(X1,Y1,Z1,X2,Y2,Z2,DD,CGD,SGD,CTH,STH,CPH,SPH,
     2 GAM,ETA,ET1,DUM,EP1,DUM)
C     ETH=ETH+W*ET1*COS(XK*(WNT/2.0-(IS-1)*HH1))
C     EPH=EPH+W*EP1*COS(XK*(WNT/2.0-(IS-1)*HH1))
C     PF=PF+COS(XK*(WNT/2.0-(IS-1)*HH1))
      ETH=ETH+W*ET1
      EPH=EPH+W*EP1
      FF=FF+W
 10   CONTINUE
      ETH=IN12*ETH/FF
      EPH=IN12*EPH/FF
      RETURN
      END
```

APPENDIX 38

SUBROUTINE SURMFF

```
      SUBROUTINE SURMFF(X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,I12,TH.PH,
     & ETH,EPH,WVL)
      COMPLEX ETH,EPH.EX,EY,EZ,EXP,EYP,EZP
      REAL L1.L2.L3 M1,M2,M3,N1,N2,N3
      PI=3.14159
C TRANSFORM TO STANDARD COORD SYSTEM
      W=.5*SQRT((X3-X2)**2+(Y3-Y2)**2+(Z3-Z2)**2)
      H=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
      X0=(X3+X1)/2.
      Y0=(Y3+Y1)/2.
      Z0=(Z3+Z1)/2.
      L2=(X2-X3)/(2.*W)
      M2=(Y2-Y3)/(2.*W)
      N2=(Z2-Z3)/(2.*W)
      L3=(X2-X1)/H
      M3=(Y2-Y1)/H
      N3=(Z2-Z1)/H
      L1=M2*N3-M3*N2
      M1=L3*N2-L2*N3
      N1=L2*M3-L3*M2
      X=SIN(TH)*COS(PH)
      Y=SIN(TH)*SIN(PH)
      Z=COS(TH)
      XP=L1*X+M1*Y+N1*Z
      YP=L2*X+M2*Y+N2*Z
      ZP=L3*X+M3*Y+N3*Z
      THP=ACOS(.999999*ZP/SQRT(XP*XP+YP*YP+ZP*ZP))
      PHP=0.0
      IF(XP**2+YP**2.GT.0.0)PHP=ATAN2(YP,XP)
      Z2P=H/2.
C COMPUTE FOR FIELD OF MONOPOLE IN STSNDARD COORD SYSTEM
      ETA0=376.7
      XK=2.*PI/WVL
      ETH=(0.,0.)
      EPH=(0.,0.)
      STH=SIN(THP)
      IF (ABS(STH) .LT. .001) RETURN
      CTH=COS(THP)
      SKW=SIN(XK*W)
      CKW=COS(XK*W)
      SPH=SIN(PHP)
      CPH=COS(PHP)
      SKH=SIN(XK*H)
      CKH=COS(XK*H)
      ETH=(0.,1.)*ETA0/(4.*PI)
      IF (ABS(CTH).LT..001 .AND. ABS(CPH).LT..001) GOTO 10
C COMPUTE NORMAL FORM OF ETH
      ETH=ETH*(SKW*COS(XK*W*STH*SPH)-(STH*SPH*CKW*SIN(XK*W*STH*SPH)))
      ETH=ETH/(SKH*SKW*(1.-STH*STH*SPH*SPH)*STH)
      ETH=ETH*CEXP((0.,1.)*XK*Z2P*CTH)
      ETH=ETH*(CEXP((0.,-1.)*XK*H*CTH)*((0.,-1.)*CTH*SKH
     & -CKH)+1.)
      GOTO 20
10    ETH=ETH*(XK*W+CKW*SKW)*(1.-CKH)/(2.*SKH*SKW)
C TRANSFORM TO ORIGINAL SYSTEM
20    ETH=ETH*CEXP((0.,1.)*XK*(X0*X+Y0*Y+Z0*Z))
      EXP=ETH*CTH*CPH
      EYP=ETH*CTH*SPH
      EZP=-ETH*STH
      EX=(L1*EXP+L2*EYP+L3*EZP)*I12
      EY=(M1*EXP+M2*EYP+M3*EZP)*I12
      EZ=(N1*EXP+N2*EYP+N3*EZP)*I12
      ETH=EX*COS(TH)*COS(PH)+EY*COS(TH)*SIN(PH)-EZ*SIN(TH)
      EPH=-EX*SIN(PH)+EY*COS(PH)
      RETURN
      END
```

APPENDIX 39

SUBROUTINE DSKFF

```
      SUBROUTINE DSKFF(X0,Y0,Z0.X1,Y1,Z1,X2,Y2,Z2,TH,PH,
     & A,B,WVL,ETH,EPH)
      COMPLEX ETH,EPH,EX,EY,EZ,EXP,EYP,EZP
      REAL L1,L2,L3,M1.M2,M3,N1,N2.N3
      PI=3.14159265
      XK=2.*PI/WVL
      ETA0=376.7
      XKBA=XK*(B-A)
C TRANSFORM TO STANDARD SYSTEM
      D=SQRT((X2-X0)**2+(Y2-Y0)**2+(Z2-Z0)**2)
      L1=(X2-X0)/D
      M1=(Y2-Y0)/D
      N1=(Z2-Z0)/D
      L3=(Y1-Y0)*(Z2-Z0)-(Y2-Y0)*(Z1-Z0)
      M3=(Z1-Z0)*(X2-X0)-(Z2-Z0)*(X1-X0)
      N3=(X1-X0)*(Y2-Y0)-(X2-X0)*(Y1-Y0)
      R=SQRT(L3*L3+M3*M3+N3*N3)
      L3=L3/R
      M3=M3/R
      N3=N3/R
      L2=M3*N1-N3*M1
      M2=L1*N3-N1*L3
      N2=L3*M1-L1*M3
      X=SIN(TH)*COS(PH)
      Y=SIN(TH)*SIN(PH)
      Z=COS(TH)
      XP=L1*X+M1*Y+N1*Z
      YP=L2*X+M2*Y+N2*Z
      ZP=L3*X+M3*Y+N3*Z
      THP=ACOS(.9999*ZP/SQRT(XP*XP+YP*YP+ZP*ZP))
      PHP=ATAN2(YP+.00000001,XP+.00000001)
C COMPUTE FAR FIELD OF DISK IN STANDARD FORM
      F=B-SIN(XKBA)/XK-A*COS(XKBA)
      ETH=(-1.,0.)*COS(THP)*SIN(THP)*F/(8.*PI*SIN(XKBA))
      ETH=ETH*ETA0*XK
C TRANSFORM TO ORIGINAL SYSTEM
      ETH=ETH*CEXP((0.,1.)*XK*(X0*X+Y0*Y+Z0*Z))
      EXP=ETH*COS(THP)*COS(PHP)
      EYP=ETH*COS(THP)*SIN(PHP)
      EZP=-ETH*SIN(THP)
      EX=L1*EXP+L2*EYP+L3*EZP
      EY=M1*EXP+M2*EYP+M3*EZP
      EZ=N1*EXP+N2*EYP+N3*EZP
      ETH=EX*COS(TH)*COS(PH)+EY*COS(TH)*SIN(PH)-EZ*SIN(TH)
      EPH=-EX*SIN(PH)+EY*COS(PH)
      RETURN
      END
```

APPENDIX 40

⊡	⦶	△	+	X	◇	⍏	X̅	Z	Y
0	1	2	3	4	5	6	7	8	9
⌧	✳	⧖	|	☆	_	|		∧	≡
10	11	12	13	14	15	16	17	18	19
→		≠	±	_		‾	∫	⊃	∨
20	21	22	23	24	25	26	27	28	29
~	≈	}	{	μ	π	Φ	θ	ψ	×
30	31	32	33	34	35	36	37	38	39
ω	λ	∝	δ	∈	η			Σ	÷
40	41	42	43	44	45	46	47	48	49
≤	≥	Δ	[	]	\	↑	√	↕	‡
50	51	52	53	54	55	56	57	58	59
←	×	↑	↓		A	B	C	D	E
60	61	62	63	64	65	66	67	68	69
F	G	H	I	¢	.	<	(	+	|
70	71	72	73	74	75	76	77	78	79
&	J	K	L	M	N	Ø	P	Q	R
80	81	82	83	84	85	86	87	88	89
!	$	*	)	;	¬	-	/	S	T
90	91	92	93	94	95	96	97	98	99
U	V	W	X	Y	Z	∞	,	%	—
100	101	102	103	104	105	106	107	108	109
>	?	0	1	2	3	4	5	6	7
110	111	112	113	114	115	116	117	118	119
8	9	:	#	©	'	=	"		
120	121	122	123	124	125	126	127		